

UNIVERSIDAD AUTÓNOMA DE BAJA CALIFORNIA

Facultad de Contaduría y Administración

Facultad de Ciencias Químicas e Ingeniería

Maestría en Tecnologías de la Información y la Comunicación



**Desarrollo de la documentación del Sistema de Gestión
Académica FCA (SIGAF).**

TESIS

PARA OBTENER EL GRADO DE

MAESTRÍA EN TECNOLOGÍAS DE LA INFORMACIÓN Y LA COMUNICACIÓN

Presenta:

Espinoza Bastidas Unice Lizbeth

Bajo la dirección de:

Dra. Hilda Beatriz Ramírez Moreno

Codirección:

Dra. Margarita Ramírez Ramírez

Tijuana, Baja California, México

Octubre 2015

Mi dedicatorio

Este trabajo de tesis se lo dedico a Dios quien me guía y fortalece para enfrentar los retos que se me presentan, a mis padres quienes por el solo de hecho de darme la vida merecen todo mi amor, respeto y admiración, a mis hermanos y sobrinos que adoro con todo mi corazón y a mi novio a quien amo; todos en conjunto complementan mi felicidad, sin ellos no hubiera sido posible este logro.

Agradecimientos

Agradezco primeramente a Dios por permitirme cursar y culminar esta etapa de mi vida profesional, a mis padres que son la base de mi persona, mi fortaleza, mi guía, quienes siempre están ahí para apoyarme, gracias por todo su amor, su comprensión, su ayuda, gracias por existir y dejarme ser parte de su vida; agradezco a mis hermanos que también son parte fundamental, por su cariño, su apoyo, su compañía, por estar siempre en los momentos hermosos y sobre todo en los difíciles porque al igual que mis padres siempre están cuando los necesito. Agradezco a mi novio quien me ha apoyado siempre y me alentó para tomar la decisión y enfrentar el reto que implica hacer una maestría, gracias por su amor y comprensión.

Agradezco también al equipo de trabajo SIGAF por formar parte importante para que esta tesis se pudiera realizar, pero sobre todo, agradezco a mis amigos Cynthia e Iván porque ellos fueron un pilar muy fuerte para mí durante toda la maestría, gracias por su amistad y apoyo.

Una vez más doy gracias a todos los maestros que me impartieron clases, su conocimiento y experiencia fueron de gran importancia, al comité de la maestría por su gran trabajo para que se pudiera lograr esto y a Conacyt porque sin su apoyo económico esta maestría habría sido difícil de solventar.

Espinoza Bastidas Unice Lizbeth

Resumen

Este trabajo se enfoca en el desarrollo de la documentación de los módulos login y usuarios, plan de estudios, carga académica y disponibilidad docente del Sistema de Gestión Académica de la Facultad de Contaduría y Administración (FCA), se hace énfasis sobre los elementos que se deben tomar en cuenta para realizar documentación de calidad, dentro de estos se considera el uso de una metodología, emplear estándares que permitan establecer las bases para que cualquier persona pueda darle al documento la interpretación correcta, aplicar características de redacción y conocer los tipos de documentos. La documentación es fundamental para cualquier empresa o institución puesto que permite fijar las pautas o lineamientos dentro de la misma así como establecer organización, planificación y control en los procesos o actividades que se llevan a cabo, esta pasa a ser parte de los activos pues en ella quedan plasmadas las mejores prácticas de la organización. Se emplea Métrica V3 como metodología para estructurar el documento de desarrollo del sistema y la metodología SCRUM para el desarrollo del proyecto; en lo que respecta a la presentación del manual de usuario, se implementará sobre la interfaz gráfica del sistema como una sección de ayuda lo que permitirá brindarle al usuario un sistema completo que le posibilite el manejo de la información de manera confiable, fácil y rápida.

Palabras clave: Documentación, manuales, estándares, sistemas, metodología.

Índice general

Capítulo I. Introducción

1. Introducción	2
1.1. Antecedentes	3
1.2. Definición del problema.....	9
1.3. Justificación.....	10
1.4. Objetivos	11
1.4.1. Objetivo general.....	11
1.4.2. Objetivos específicos	11
1.5. Alcance del proyecto	12

Capítulo II. Marco teórico

2. Marco teórico	15
2.1 Documentación de sistemas.	15
2.1.1 Características de la documentación de sistemas.	16
2.1.2 Presentación de la documentación.	17
2.1.3 Tipos de documentación	18
2.1.3.1 Manual Administrativo.	18
2.1.3.2 Manual de Usuario.....	19
2.1.3.3 Manual Técnico.	20
2.1.3.4 Manual de Instalación.....	21
2.1.4 Estándares para documentar sistemas.	21
2.1.4.1 Estándar para Especificación de Requerimientos de Software.....	22
2.1.4.2 Estándar UML.....	24
2.1.4.3 Estándar Web	25

2.1.4.4 Estándares de calidad del software	26
2.2 Desarrollo de Interfaz Gráfica de Usuario para consulta de documentación.	29
2.2.1 Ingeniería Web.....	29
2.2.1.1 Ventajas de usar las aplicaciones web.....	29
2.2.1.2 Lenguajes de diseño y programación web.....	31

Capítulo III. Metodología

3. Metodología	36
3.1 Planificación de Sistemas de Información.....	37
3.2 Desarrollo de Sistemas de Información	39

Capítulo IV. Desarrollo

4. Desarrollo.....	42
4.1 Creación de la interfaz gráfica para la consulta del manual de usuario	42
4.1.1 Boceto para la sección de ayuda.....	42
4.1.2 Ubicación de la ayuda	43
4.1.3 Pantalla inicial de la ayuda.....	44
4.1.3.1 Creación de la pantalla inicial.	45
4.1.4 Pantalla para cada uno de los módulos.	48
4.1.4.1 Creación de la plantilla para cada uno de los módulos.	50
4.2 Documentación del Sistema	53
4.2.1 Control de cambios	56
4. 2. 2 Plan de estudios.....	58
4.2.2.1 Fase de análisis	58
4.2.2.2 Fase de diseño del sistema (Interfaz Gráfica de Usuario y BD)	61
4.2.2.3 Fase de construcción del sistema (programación)	63
4.2.2.4 Fase de pruebas e implantación del sistema.....	65

4.2.3 Carga académica	67
4.2.4 Disponibilidad docente	67
4.2.5 Login y usuarios.....	67
4.2.6 Manual de usuario	68

Capítulo V. Resultados

5. Resultados	70
---------------------	----

Capítulo VI. Conclusiones y recomendaciones

6. Conclusiones y recomendaciones	76
Bibliografía.....	78
Anexos	81

Índice de figuras

Figura 1. Proceso de desarrollo de SCRUM (Itequia, 2014)	28
Figura 2. Pantalla principal de Base Camp.	54
Figura 3. Lista de los integrantes del proyecto participantes en Base Camp.....	55
Figura 4. Contenido de las minutas.....	56
Figura 5. Pantalla principal de ayuda.	42
Figura 6. Pantalla con las opciones principales de ayuda.....	43
Figura 7. Pantalla principal del Sistema.	43
Figura 8. Ubicación del manual de ayuda dentro del plan de estudios.	44
Figura 9. Pantalla inicial de la ayuda.	44
Figura 10. Ayuda del módulo plan de estudios.	48
Figure 11. Plantilla para cada uno de los módulos.....	50
Figura 12. Estructura general de la plantilla de ayuda para cada uno de los módulos.	51
Figure 13. Referencia de las hojas de estilo.	52
Figura 14. Contenido de la fase de análisis.	59
Figura 15. Contenido de la fase de diseño.	62
Figura 16. Contenido de la fase de construcción (programación).	63
Figura 17. Contenido de la fase de implantación y pruebas.....	65
Figura 18. Contenido del manual de usuario.....	68
Figura 19. Pantalla principal de la sección de ayuda.....	72
Figura 20. Interfaz de ayuda del módulo plan de estudios.	73
Figura 21. Interfaz de ayuda del módulo de carga académica.....	73
Figura 22. Interfaz de ayuda del módulo disponibilidad docente.	74

Índice de tablas

Tabla 1. Elementos que componen la pantalla inicial de la ayuda..... 45

Tabla 2. Elementos que integran el contenido de la plantilla. 52

Capítulo I. Introducción

1. Introducción

El presente trabajo se centra sobre la documentación de los módulos login y usuarios, plan de estudios, carga académica y disponibilidad docente los cuales forman parte del Sistema de Gestión Académica FCA (SIGAF), dicho sistema se enfoca en la gestión de la información académica de la Facultad de Contaduría y Administración (FCA) de la Universidad Autónoma de Baja California (UABC). Hoy en día en la FCA se emplea un sistema para la creación del horario de los docentes, sin embargo, se presenta la necesidad de mejorarlo debido a que este no cuenta con un diseño apegado a la necesidad del usuario, principalmente en la consulta de información; no se dispone del código fuente ni de documentos o manuales que permitan familiarizarse con su proceso de trabajo para efectuar posibles actualizaciones o mantenimiento, para dar solución a esta situación se desarrolló un sistema al cual se le agregaron funcionalidades que permiten la inserción, eliminación, consulta y actualización de la información en un ambiente interactivo con su respectiva documentación lo que le brinda al usuario la posibilidad de tomar decisiones oportunas.

El funcionamiento del sistema SIGAF queda sustentado en la documentación del desarrollo del sistema y los manuales de usuario y técnico; a medida que se desarrolla el marco teórico de este trabajo se contemplan los elementos que engloba una documentación de calidad como lo es el saber qué tipo de documentos es necesario generar en un desarrollo de software, identificar a que usuarios van dirigidos dichos documentos, que características deben considerarse para su redacción, definir estándares, emplear una metodología que sirva como guía para documentar, en este caso se empleó Métrica 3 la cual contempla las actividades a realizar en las etapas de

desarrollo del software; para darle seguimiento al desarrollo del software se empleó la metodología SCRUM, al iniciar el proyecto se definieron una serie de Sprints con sus respectivas tareas.

1.1. Antecedentes

Actualmente la Universidad Autónoma de Baja California (UABC), campus Tijuana, en la Facultad de Contaduría y Administración posee un Sistema de Administración de Horarios del docente para realizar las funciones básicas que se requieren y suplir las necesidades que la facultad tiene en el manejo de los horarios, dicho sistema es aplicado para las carreras a nivel licenciatura (Administración de Empresas, Informática, Negocios Internacionales y Contaduría), este proceso se realiza semestralmente y maneja aproximadamente a 300 maestros, 120 grupos, un total aproximado de 4000 alumnos y la formación de turnos matutinos, interturnos y nocturnos.

El proceso de elaboración del horario del docente requiere que este dado de alta el plan de estudio en curso, el cual puede ser registrado aproximadamente cada cinco años por las analistas o el subdirector de la facultad; la carga académica la realizan los coordinadores en formatos de Excel y la pasan a subdirección donde se captura en el actual sistema; la disponibilidad del docente, es un procedimiento que realiza cada maestro dos meses antes de concluir el semestre, se les entrega físicamente o se les envía por correo electrónico para su llenado; las actividades que se mencionan anteriormente son introducidas de forma manual al sistema por el personal de la subdirección(analistas o subdirector) y solo conducen a la creación del horario del docente.

La formación del horario del docente se hace en base a una copia de la base de datos del semestre anterior y de ahí se parte para la creación de la nueva asignación, una observación importante la cual se debe considerar es que los grupos tengan clases continuas evitándose horarios libres entre clases, el sistema permite realizar estos ajustes, sin embargo, el manejo de la información como se realiza actualmente no ayuda a la toma de decisiones y se emplea mucho tiempo y esfuerzo en realizar estas actividades. Por otro parte, algunas otras facultades de la universidad han tomado este sistema, viéndose en la dificultad para aprender su funcionamiento ya que no se tiene la documentación que permita familiarizarse con su proceso de trabajo.

En la actualidad existen diversos sistemas comerciales enfocados en la creación de los horarios, uno de ellos es Untis Express, este es un software manejado a través del entorno Windows que permite generar los horarios de acuerdo a ciertos parámetros que puede definir el usuario, cuenta con un asistente que mejora el manejo de la introducción de datos básicos para que su funcionamiento sea más amigable. Untis Express está definido por un algoritmo que no solo calcula un resultado de horarios sino múltiples opciones que le brindan más posibilidades de elección al usuario, permite imprimir el horario generado, planificar sustituciones en el caso en que algún maestro sea dado de baja o no se encuentre en condiciones para impartir clase, posee un control de asistencias de maestros lo cual permite tomar decisiones al personal encargado al momento de generar el horario, otra de las ventajas de este software es que cuenta con ayuda Online para guiar a los usuarios por las características del programa de forma precisa; la ayuda está dividida en tres secciones: el contenido, el índice y la búsqueda,

esta clasificación le permite al usuario consultar la información del sistema de la manera que considere adecuada.

De acuerdo al portal de difusión de Untis Express, desde 1984 este sistema ha sido empleado por los centros escolares en la mayoría de los países Europeos con éxito en Alemania, Francia, Inglaterra, Italia, Bélgica, Finlandia, Suecia, Grecia, existe también desde 1988 una versión en español y catalán (Untis Express, 2013).

Kronowin es otra aplicación que permite obtener los horarios óptimos de un centro escolar, ya sean sencillos o complejos, en función de la magnitud del centro educativo. Por medio de este programa se facilita la tarea de cuadrar los horarios para los docentes, grupos o clases y aulas, permite ajustar la estructura del horario, filtrar cada horario por asignatura, ofrece cabeceras modificables para no tener que rehacer todo un horario por un ligero cambio, cuenta con un módulo donde cada departamento solicita la pre asignación de materias por profesor, también dispone de un manual online que enseña cómo realizar cada paso en la creación de los horarios. Kronowin es una herramienta que ofrece ventajas a las instituciones educativas al auxiliarlos en la creación de sus horarios, sin embargo, esta herramienta se encuentra restringida a 10 asignaturas, 10 profesores y 10 grupos lo que lo hace limitado para instituciones grandes (Braña, 2013).

AGH/iX es un producto estándar con un gran número de usuarios y con más de 15 años de desarrollo continuo elaborado por la empresa Centro de Desarrollo Académico de Catalonia. Este programa permite al usuario crear y gestionar los horarios docentes en centros de enseñanza superior, aulas, profesores, grupos de alumnos y asignaturas de

los grados y titulaciones de modo integrado y eficaz (Catalonia Software Solutions, 2013).

Como se menciona en los tres sistemas descritos anteriormente, existe variedad de software que permiten abatir el problema de la creación de horarios, sin embargo, en cada institución educativa se toman diferentes criterios para la generación de estos, esta es una razón por la cual la Universidad Autónoma de Baja California, en la Facultad de Contaduría y Administración no ha implementado un software de este tipo, la institución requiere de un software que se desarrolle a la medida de sus necesidades, que opere vía web.

Un elemento clave de un sistema de software es la documentación, la cual debe estar disponible y ser fácil de consultar, como se mencionó hay sistemas de control de horarios que traen integrado en la interfaz de usuario una guía de ayuda, lo que le permite al usuario consultar la información del uso adecuado de este de una manera amigable y rápida.

Otra de las vertientes que engloba la necesidad de un software de control de horarios en la Facultad de Contaduría y Administración es que cuente con la documentación que respalde su funcionamiento, escalabilidad y mantenimiento, dada la importancia que implica documentar ya que esta brinda toda la información necesaria para conocer el funcionamiento correcto de los procesos que se llevaron a cabo para realizar el sistema, permite que el software sea escalable, esto es que tenga posibilidades de añadirle nuevas funcionalidades o en su defecto actualizarlo, también posibilita su mantenimiento, brinda los lineamientos para que se realicen auditorías de este en caso

de ser necesario ya que estas se basan específicamente en los documentos, de igual forma ayuda a eliminar la posible dependencia que se pueda formar entre el proyecto realizado y la persona que se encarga de usarlo y además, proporciona prestigio a la empresa ya que en la documentación se plasma el capital intelectual que esta posee, es decir el “saber hacer algo bien hecho” que se ha adquirido con la experiencia y por lo cual las personas están dispuestas a pagar, si no se preserva adecuadamente ese conocimiento difícilmente logrado, se podría perder.

Para que toda aplicación tecnológica y todo servicio informático quede adecuadamente documentado, es necesario que se cuente con personal que realice esta actividad (puede ser personal interno o un proveedor) que entregue dicha documentación obtenida a través del desarrollo del proyecto, de manera que otras personas relacionadas o autorizadas por la empresa accedan a los conocimientos necesarios para corregir errores, hacer ajustes, o para lo que haya necesidad. La oposición de los técnicos a la realización del proceso de documentación, se basa principalmente en el gran trabajo que esto implica y en el mayor plazo de entrega del proyecto, sin embargo, estas justificaciones no son posibles de sustentar en la realidad, ya que el mayor costo en que se incurre para documentar se recupera con creces en el futuro, especialmente cuando el usuario se ve enfrentado a un error del sistema o se le presenta la necesidad de darle mantenimiento correctivo por mal uso.

A medida que avanza la elaboración del proyecto de software se debe documentar cada una de sus etapas, lo cual permitirá conocer el funcionamiento de los procesos que se llevaron a cabo para realizar el sistema, en la planificación deben quedar las bases documentales donde se defina claramente las tareas, responsabilidades de cada

miembro del equipo y las fechas clave para su realización lo cual evitará malos entendidos, retrasos y descoordinación. La documentación de un sistema de software no se puede concebir como una actividad aislada, sino como el resultado de la aplicación de diversas técnicas, elementos y metodologías para su elaboración.

Es importante mencionar que la documentación durante el desarrollo de un proyecto permite estandarizar los procesos, al realizar esto se asegura que las personas se apeguen a lo establecido y realicen los procesos tal y como están escritos; proporciona planificación organizacional, al documentar elementos importantes como acciones, recursos y tiempos, esto guía a las personas a cargo hacia el logro de determinados objetivos o metas; también por medio de esta se logra el control de actividades realizadas, tener documentación brinda evidencia o soporte de la realización de determinadas actividades y contribuye al mejoramiento de procesos.

1.2. Definición del problema

El problema se presenta en la Facultad de Contaduría y Administración (FCA) en la Universidad Autónoma de Baja California (UABC), campus Tijuana, a nivel licenciatura, y surge por la necesidad de mejorar el manejo del control de horarios de los docentes, actualmente la FCA cuenta con un sistema para realizar este trabajo el cual fue desarrollado hace aproximadamente 15 años, sin embargo, este sistema se encuentra limitado, no proporciona la información que el usuario desea obtener para tomar decisiones oportunas, gran parte de la información se introduce manualmente y no ha sido actualizado desde su implementación ya que no se tiene ni el código fuente, ni la documentación para darle mantenimiento.

Dada esta problemática se desarrollará un sistema vía web que permita gestionar la información de los usuarios, los planes de estudio, la carga académica y la disponibilidad docente permitiéndole al usuario trabajar de una forma óptima ahorrándole tiempo y esfuerzo en su labor, dicho sistema deberá contar con la documentación completa del proyecto, el manual de usuario y técnico que respalden su funcionamiento y las posibles actualizaciones que pudieran surgir en un futuro. La documentación se presentará como parte de la interfaz gráfica del sistema brindándole al usuario una forma más rápida y amigable de interactuar con esta. El sistema tiene roles definidos para que el control de la información sea confiable, en base al rol será el acceso a la información que el usuario podrá consultar y/o manipular. La problemática planteada anteriormente se pretende resolver mediante el desarrollo y la documentación del sistema en un tiempo estimado de 1 año 3 meses, finalizará en junio del 2015 con la implementación del Sistema de Gestión Académica y la presentación de tesis.

1.3. Justificación

La importancia de documentar un sistema de software radica en:

- Mantener controles internos en la organización que permitan una mejor gestión y desempeño por parte del personal en cada área, esto contribuye a disminuir costos de operación.
- Conocer los procesos que se realizan dentro de la organización para adaptarse rápidamente a los cambios que involucren ampliaciones o mantenimiento que esta pudiera llegar a necesitar.
- Cuando se producen contratiempos graves, el tiempo de restauración de los sistemas y procesos es uno de los elementos más críticos, la documentación acorta los periodos de recuperación y contribuye a corregir los errores de modo más eficiente.
- Desde el punto de vista del usuario, el manual permite conocer la utilidad y el manejo apropiado del sistema.
- Las herramientas digitales interactivas permiten comunicar visualmente el contenido de la documentación típica de manera más eficiente y amigable para el usuario.

1.4. Objetivos

1.4.1. Objetivo general

Elaborar la documentación de los módulos plan de estudios, carga académica, disponibilidad docente y login y usuarios, los cuales forman parte del Sistema de Gestión Académica FCA (SIGAF) y el desarrollo de la Interfaz Gráfica de Usuario (GUI) para la consulta de dicha documentación.

1.4.2. Objetivos específicos

1. Documentar la información originada de los procesos o actividades realizadas dentro del desarrollo del software.
2. Elaborar el manual técnico y de usuario.
3. Desarrollar una GUI que permita consultar el manual de usuario en base al acceso que este tenga sobre el sistema.

1.5. Alcance del proyecto

Se contempla que los usuarios del sistema SIGAF tengan acceso a la información a través de la interfaz web, el desarrollo y la interfaz en si quedarán debidamente documentados.

Alcances:

1. Documentación de los módulos que componen el Sistema de Gestión Académica FCA (SIGAF), conforme se realice el análisis, diseño de interfaces (frontend), diseño de base de datos, programación (backend), pruebas e implementación de cada uno de ellos. Los módulos en cuestión son:
 - **Login y usuarios:** esta sección está compuesta por la administración de los catálogos (país, estado, ciudad, unidad académica, categoría, campus) y las funcionalidades de registro, consulta, modificación y eliminación de usuarios así como la posibilidad de cambiar contraseña y recuperar en caso de no recordarla, los usuarios tendrán acceso al sistema de acuerdo al tipo de usuario que inicie sesión.
 - **Plan de estudios:** comprende las interfaces para la administración de los catálogos (seriación, carácter, etapas, nivel programa, núm. de plan, coordinación) y las funcionalidades de registro, consulta, eliminación y actualización del plan de estudios.
 - **Carga académica:** este módulo contiene los catálogos (periodo, grupo) y las funcionalidades de registro inicial el cual se utiliza cuando se registra una carga académica por primera vez, el registro subsecuente permite crear una copia de la carga anterior registrada y guardarla con el periodo en curso, en

dicha carga se pueden realizar las modificaciones pertinentes, como asignar las unidades de aprendizaje a los grupos, quitar las que no estén contempladas para la carga académica actual, borrar la carga completa y consultar la carga académica.

- **Disponibilidad docente:** el módulo de disponibilidad docente engloba los catálogos (empresa, puesto, carrera, escuela, cursos) y le permite al usuario subir su foto y los archivos que comprueben su preparación profesional y congresos o talleres asistidos o próximos a cursar así como el registro y consulta de la disponibilidad docente.

2. Elaboración del manual técnico y de usuario del sistema.

3. Desarrollo de la interfaz que permita consultar la documentación del manual de usuario, en cada módulo del sistema estará disponible la interfaz con la documentación correspondiente.

Limitaciones:

Limitaciones físicas: el sistema SIGAF está considerado para utilizarse en la Facultad de Contaduría y Administración UABC campus Tijuana en el nivel de licenciaturas (Licenciado en Contaduría, Licenciado en Administración de Empresas, Licenciado en Informática y Licenciado en Negocios Internacionales), sin embargo, en un futuro se pretende que pueda ser parte de otras facultades dentro de la universidad.

Capítulo II. Marco teórico

2. Marco teórico

2.1 Documentación de sistemas.

Documentar los sistemas implica reunir información relacionada a este que explique la parte lógica del sistema, sus características técnicas y funcionales, los diagramas de flujo, los programas, capacidades del sistema y términos y condiciones de uso, la documentación de sistemas en sí, es el conjunto de información que nos dice qué hacen los sistemas, cómo lo hacen y para quién lo hacen, con la finalidad de que puedan ser usados, operados y/o modificados.

Los principales objetivos del proceso de documentación son:

- Registrar la información producida por cualquier proceso o actividad del ciclo de vida del software.
- Gestionar los documentos necesarios para todas las personas involucradas en el proceso de software, tales como directores, ingenieros, personal de desarrollo, usuarios del sistema (Pérez, 2009).

Existen razones válidas para escribir documentación en un proyecto relacionado con el software:

- Es un requisito de un proyecto comercial, donde el cliente solicita que se documenten distintos componentes o fases del proyecto.
- Se emplea para definir un modelo contractual que especifica cómo interactúan dos subsistemas diferentes; este tipo de modelos suelen ser necesarios cuando hay más de un grupo de trabajo en distintos componentes de un mismo sistema

(por ejemplo cuando una aplicación necesita interactuar con base de datos externas o aplicaciones preexistentes).

- Para permitir la comunicación con equipos externos; cuando se trabaja con personas geográficamente dispersas debe existir un medio para comunicarse y la documentación compartida es en muchas ocasiones parte de la solución, pero siempre combinada con otros canales como discusiones cara a cara, teleconferencias, correo electrónico o el uso de herramientas colaborativas.
- Para pensar algo en profundidad, mucha gente escribe documentación para aclarar cosas discutidas en un grupo o para aumentar su comprensión en un determinado problema, el escribir ideas en un papel puede ayudar a consolidarlas (Talens-Oliag, 2013).

2.1.1 Características de la documentación de sistemas.

A medida que avanza el desarrollo del sistema se debe documentar todo su proceso, para que el contenido del documento le proporcione la ayuda que el lector requiere, el documento debe contener las siguientes características básicas:

- Claridad: la redacción del documento será un texto fácil de entender.
- Concisión: el escrito debe ser breve, debe contener específicamente la información que al usuario le concierne.
- Precisión: el contenido debe tener un solo sentido, no debe contener ambigüedades, es decir, no debe provocar que el usuario le dé diferente interpretación a la información proporcionada.

- Sencillez: en la redacción se utiliza palabras simples, sin rebuscamiento, no tecnicismos excesivos y sin expresiones coloquiales.
- Coherencia: organizar el texto a través de estructuras lógicas, manejo de párrafos y de capítulos, o a través del orden cronológico o espacial, y de otros factores de acuerdo a cada trabajo.

2.1.2 Presentación de la documentación.

La documentación en línea, según Brun (1998), consiste en instrucciones escritas, compartidas mediante formato electrónico, que permiten obtener las máximas prestaciones de las aplicaciones de software.

La documentación en línea establece reglas que permiten estandarizar los procedimientos de trabajo soportados por las aplicaciones informáticas. Los siguientes puntos muestran las ventajas que ofrece el documento en formato digital:

- Mejores mecanismos de acceso a la información mediante búsquedas en texto completo, por palabras clave, tablas de contenidos jerárquicas, desplegables, entre otros.
- Mínimos costos de distribución.
- Facilidad para personalizar y parametrizar la documentación mediante anotaciones, marcas de lectura, -bookmark- e incluso la propia edición y modificación de texto.
- Mayor rapidez y flexibilidad en la distribución de actualizaciones del producto, mejoras y errores corregidos.
- Se requiere de menos tiempo para publicar la información.

La principal característica de un sistema de documentación en línea para aplicaciones de software y uno de los factores críticos para poder designar con este nombre a un sistema, es la vinculación del contenido informativo con la presentación de la aplicación donde se describa claramente el manejo de esta, de forma que, en cualquier momento, el usuario pueda obtener la información que explica la parte de la aplicación con la que se trabaja.

Otra forma de presentar la documentación es de forma impresa, esta comprende los documentos escritos durante el desarrollo del software que se entregan al usuario, específicamente lo que concierne a los manuales, el resto de los documentos que se pueden generar durante el desarrollo del sistema sirven para la gestión del proyecto, dichos documentos son elaborados mediante alguna herramienta de procesamiento de texto, se entregan de forma impresa (en papel), puede ser en forma de libro o en hojas sencillas, esto depende del proyecto y de las necesidades de los involucrados.

2.1.3 Tipos de documentación

En la documentación de un proyecto de software debe existir como requisito mínimo un manual administrativo, manual de usuario, manual técnico y el manual de instalación.

2.1.3.1 Manual Administrativo.

Está representado por todos los documentos de manejo gerencial de un proyecto según el autor Caéz (2013). Este manual sirve como medio de comunicación y coordinación, permite registrar, transmitir en forma ordenada y sistemática la información de la

organización; los documentos que se contemplan dentro de este manual son los siguientes:

- Contratos
- Comunicaciones
- Actas
- Cronogramas
- Requerimientos básicos del usuario final.

2.1.3.2 Manual de Usuario.

Según Álvarez (2006), un manual de usuario contiene la información necesaria para que los usuarios utilicen correctamente una aplicación, este documento se hace desde la guía técnica pero se suprimen los tecnicismos y se presenta de forma que sea entendible para el usuario que no sea experto en el área. Un punto que se debe considerar es que en su contenido no debe haber referencias a ningún apartado de la guía técnica y en el caso de que se haga uso de algún tecnicismo debe ir acompañado de un glosario al final de la misma para su fácil comprensión.

En la siguiente lista se muestran algunas guías basadas en estudios prácticos y empíricos que sirven como base para la creación del manual de usuario (Shneiderman, 2006.):

- Elegir una aproximación orientada a la acción.
 - Proporcionar una oportunidad inmediata para actuar.
 - Estimular y dar soporte a la exploración y la innovación.
 - Respetar la integridad de la actividad del usuario.
 - Mostrar numerosos ejemplos

- Dar soporte al reconocimiento y recuperación de errores.
 - Evitar errores siempre que sea posible.
 - Proporcionar información sobre errores cuando las acciones son propensas a errores o cuando la corrección es difícil.
 - Proporcionar información sobre errores que dé soporte a la detección, diagnóstico y corrección.
 - Proporcionar información sobre errores inmediata.
- Dar soporte a una lectura para hacer, estudiar y localizar.
 - Brevedad; no explicarlo todo en detalle.
 - Proporcionar una tabla de contenido, un índice y un glosario.
 - Mantener un estilo de escritura simple y limpia.
 - Ofrecer una conclusión en cada capítulo.

2.1.3.3 Manual Técnico.

En la guía técnica o manual técnico, se reflejan el diseño del proyecto, la codificación de la aplicación y las pruebas realizadas para su correcto funcionamiento. Generalmente este documento está diseñado para los programadores o para personas con conocimientos de informática. El principal objetivo de este es facilitar el desarrollo, corrección y futuro mantenimiento de la aplicación de una forma rápida y fácil. Esta guía está compuesta por tres apartados claramente diferenciados:

- Cuaderno de carga: Es donde queda reflejada la solución o diseño de la aplicación. Esta parte de la guía es únicamente destinada a los programadores. Debe estar realizado de tal forma que permita la división del trabajo.

- Programa fuente: Es donde se incluye la codificación realizada por los programadores, este documento puede contener comentarios entre las líneas del código para su mejor comprensión y puede ser de gran ayuda para el mantenimiento o desarrollo mejorado de la aplicación, su redacción debe ser clara y precisa.
- Pruebas: es el documento donde se especifican el tipo de pruebas realizadas a lo largo de todo el proyecto y los resultados obtenidos (Alvarez, 2006).

2.1.3.4 Manual de Instalación.

El manual de instalación tiene como objetivo principal servir de guía en la instalación del sistema, en primer lugar, se deben especificar los requerimientos de hardware y software necesarios para el correcto funcionamiento de este, posteriormente se describen los pasos necesarios para la configuración e instalación (es importante asociar las imágenes del sistema en cada paso para que el usuario pueda familiarizarse más rápido con el proceso), además, se deberán incluir las pruebas que se deben realizar para asegurar que la instalación se ha realizado correctamente.

2.1.4 Estándares para documentar sistemas.

Un estándar es un modelo, criterio, regla de medida o de los requisitos mínimos aceptables para la operación de productos, procesos específicos o servicios con el fin de asegurar la calidad. Los estándares señalan claramente el comportamiento esperado y deseado y son utilizados como guías para evaluar su funcionamiento y lograr el mejoramiento continuo de los servicios. Requieren ser establecidos con el fin de contar

con una referencia que permite identificar oportunamente las variaciones presentadas en el desarrollo de los procesos y aplicar las medidas correctivas necesarias. (Sánchez, 2013).

Los estándares pueden ser desarrollados por la propia empresa, por sociedades profesionales o por organismos internacionales.

Los principales estándares involucrados en el proceso de documentación son:

- IEEE 830 1998 (especificación de requerimientos del software).
- Estándar UML (lenguaje de modelado unificado)
- ISO 12207 (proceso del ciclo de vida del software).
- IEEE 1074-1998 (proceso de ciclo de vida del software).
- Estándares Web (HTML, CSS).
- ISO/IEC 15.504 (mejora de procesos de software y determinación de normas de garantía de capacidad).
- ISO 9000-3 (gestión de la calidad).
- IEEE 610.12-1998 (terminología de glosario estándar de Ingeniería de Software).
- Métrica 3 (metodología de planificación, desarrollo y mantenimiento de sistemas de información).

2.1.4.1 Estándar para Especificación de Requerimientos de Software

El Instituto de Ingenieros Eléctricos y Electrónicos(IEEE) desarrolló el estándar IEEE 830 1998 el cual está dirigido a la especificación de requerimientos para el desarrollo de software, en este se establecen una serie de características que deben tener los requerimientos (deben ser correctos, consistentes, completos, realistas, rastreables y

verificables), tipos de requerimientos (funcionales y no funcionales), lo que se debe tomar en cuenta al elaborarlos (ambiente físico, interfaces, usuarios y factores humanos, funcionalidad, documentación, datos, recursos, seguridad y aseguramiento de la calidad), así como también la estructura para la documentación de estos (Fuentes, 2011).

Según el estándar de IEEE, define un requerimiento como:

- Una condición o necesidad de un usuario para resolver un problema o alcanzar un objetivo.
- Una condición o capacidad que debe estar presente en un sistema o componentes de un sistema para satisfacer un contrato, estándar, especificación u otro documento formal.

Cada requerimiento debe expresarse de modo adecuado, ser de fácil acceso, debe ir en un orden numerado, requiere de estar acompañado de pruebas que lo verifiquen, debe ser considerado en el diseño y la programación, probarse de forma aislada y junto a otros requerimientos y validarse con las pruebas después de construirse la aplicación.

Los requerimientos pueden dividirse en requerimientos funcionales y requerimientos no funcionales:

- Los requerimientos funcionales definen las funciones que el sistema será capaz de realizar, describen las transformaciones que el sistema realiza sobre las entradas para producir salidas.
- Los requerimientos no funcionales están relacionados con características que de una u otra forma puedan limitar el sistema, como por ejemplo, el rendimiento

(en tiempo y espacio), relación con otras interfaces de usuario, fiabilidad (robustez del sistema, disponibilidad de equipo), mantenimiento, seguridad, portabilidad, estándares, entre otras.

2.1.4.2 Estándar UML

El Lenguaje Unificado de Modelado(UML) según el autor Grady Booch (2006) es un lenguaje estándar para escribir planos de software, se utiliza para visualizar, especificar, construir y documentar los artefactos de un sistema que involucre una gran cantidad de software. UML es apropiado para modelar desde sistemas de información empresariales hasta aplicaciones distribuidas basadas en la Web, e incluso para sistemas embebidos en tiempo real muy exigentes. Es un lenguaje muy expresivo, no es difícil de aprender ni de utilizar; representa solo una parte de un método de desarrollo de software, UML es independiente del proceso, aunque para utilizarlo óptimamente se debería usar en un proceso que fuese dirigido por los casos de uso, centrado en la arquitectura, iterativo e incremental.

El lenguaje proporciona un vocabulario y las reglas para combinar palabras con el objetivo de posibilitar la comunicación. Un lenguaje de modelado es un lenguaje cuyo vocabulario y reglas se centran en una representación conceptual y física de un sistema. Para sistemas que sean extensos, se requiere un lenguaje que abarque las diferentes vistas de la arquitectura del sistema conforme evoluciona a través del ciclo de vida del desarrollo de software.

El vocabulario de UML incluye tres clases de bloques básicos:

1. Elementos (estructurales, comportamiento, agrupación y anotación).
2. Relaciones (dependencia, asociación, generalización y realización).
3. Diagramas (clases, objetos, componentes, estructura compuesta, casos de uso, secuencia, comunicación, estados, actividades, despliegue, paquete, tiempos y visión global de interacciones).

2.1.4.3 Estándar Web

El W3C es un consorcio internacional que desarrolla estándares relacionados con la Web también conocidos como recomendaciones, que sirven como referencia para construir una Web accesible, interoperable y eficiente, en la que se puedan desarrollar aplicaciones cada vez más robustas.

En la creación de las recomendaciones del W3C participan sus miembros (más de 400 organizaciones, distribuidas a lo largo de todo el mundo y de diversos ámbitos: grandes empresas de hardware o software, centros investigadores, universidades, administraciones públicas, entre otros.), el equipo del W3C, expertos invitados, y cualquier usuario de la Web que quiera mostrar su opinión. Todos ellos trabajan conjuntamente a través de un proceso basado en el consenso, la neutralidad y la transparencia de la información.

Algunos de los estándares Web más conocidos y ampliamente utilizados son: HTML (Hypertext Markup Language), para definir la estructura de los documentos; XML (eXtensible Markup Language), que sirve de base para un gran número de tecnologías;

y CSS (Cascading Style Sheets), que permite asignar estilos para la representación de los documentos (World Wide Web consorcio, 2014).

2.1.4.4 Estándares de calidad del software

El estándar ISO 9000-3 señala que el control de calidad debe ser aplicado a todas las fases de la producción de software, incluido el mantenimiento y tareas posteriores a su implantación. Este estándar se relaciona con los estándares ISO 9001 y el IEEE 730.

El ámbito de aplicación de esta norma se da sobre el desarrollo de sistemas de información, el proceso del ciclo de vida y la calidad del software. El alcance de la ISO 9000-3 busca dar orientaciones en situaciones en las que se exige la demostración de capacidad de un proveedor para desarrollar, suministrar y mantener productos de software, también sugiere clases de control para la producción de software que satisfaga los requisitos establecidos (Meneses, 2012).

Algunos de los beneficios que se obtienen con la aplicación de esta norma son:

- Mejor documentación de los sistemas.
- Cambio cultural positivo.
- Incremento en la eficiencia y productividad.
- Mayor percepción de calidad.
- Se amplía la satisfacción del cliente.
- Se reducen las auditorias de calidad.
- Agiliza el tiempo de desarrollo de un sistema.

2.1.4.5 Métrica 3

Métrica 3 según el portal de administración electrónica del gobierno de España (2014) es una metodología que involucra la planificación, desarrollo y mantenimiento de un sistema de información, ofrece a las organizaciones un instrumento para sistematizar las actividades que dan soporte al ciclo de vida del software en el desarrollo de sistemas de información y un marco de gestión para asegurar que los proyectos cumplan con sus objetivos en términos de calidad, coste y tiempo.

Los principales objetivos que ayuda a alcanzar Métrica 3 son:

- Proporcionar o definir sistemas de información que ayuden a conseguir los fines de la organización mediante la definición de un marco estratégico para el desarrollo de los mismos.
- Dotar a la organización de productos software que satisfagan las necesidades de los usuarios dando una mayor importancia al análisis de requisitos.
- Mejorar la productividad de los departamentos de sistemas y tecnologías de la información y las comunicaciones al permitir una mayor capacidad de adaptación a los cambios y tener en cuenta la reutilización en la medida de lo posible.
- Facilitar la comunicación y entendimiento entre los distintos participantes en la producción de software a lo largo del ciclo de vida del proyecto, teniendo en cuenta su papel y responsabilidad, así como las necesidades de todos y cada uno de ellos.
- Facilitar la operación, mantenimiento y uso de los productos de software obtenidos.

SCRUM

SCRUM es una metodología ágil (iterativo e incremental) de gestión de proyectos que se basa en la adaptación continua a las circunstancias evolutivas del proyecto apoyándose en iteraciones cortas conocidas como Sprint, las iteraciones se pueden entender como mini proyectos o módulos dentro de los cuales se repite un proceso similar de trabajo, se deben desarrollar en un tiempo aproximado de 1 a 4 semanas máximo, a medida que se realizan las iteraciones el producto evoluciona, se le añaden nuevos objetivos o requerimientos o se mejoran los que ya fueron completados.

En Scrum se realizan entregas parciales y regulares del producto final, priorizadas por el beneficio que aportan al receptor del proyecto, está indicada especialmente para proyectos en entornos complejos, donde se necesita obtener resultados pronto, los requisitos son cambiantes o poco definidos, donde la innovación, la competitividad, la flexibilidad y la productividad son fundamentales (Manuel Amadeo González Gálvez, 2013). En la siguiente figura se muestra el proceso de desarrollo de SCRUM.

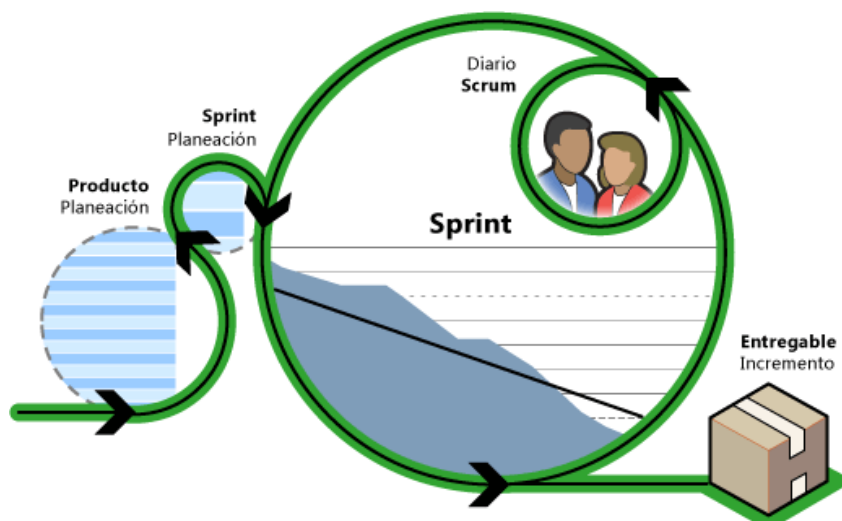


Figura 1. Proceso de desarrollo de SCRUM (Itequia, 2014)

2.2 Desarrollo de Interfaz Gráfica de Usuario (GUI) para consulta de documentación.

La documentación del manual de usuario puede ser consultada por los usuarios a través de la interfaz denominada manual y ayuda la cual está contenida dentro del sistema. A continuación se presenta información concerniente al uso de una interfaz gráfica de usuario y los elementos que se deben considerar para la creación y manejo de esta.

2.2.1 Ingeniería Web.

Las necesidades actuales de desarrollo de software están altamente influenciadas por el impacto de internet. Cada vez es más frecuente que las empresas e instituciones necesiten y dependan de aplicaciones basadas en la web, lo cual requiere que los desarrolladores tengan un amplio conocimiento de las tecnologías de vanguardia, además es fundamental conocer las metodologías y herramientas que permitan analizar, planear, modelar e implementar un sistema de calidad que responda a los requerimientos del usuario, tiempo y costos y que permita llevar a cabo el mantenimiento, crecimiento y evolución de las aplicaciones. Estas consideraciones se encuentran en la Ingeniería Web, que promueve una forma de trabajo disciplinada y sistemática y el uso de herramientas que faciliten el desarrollo, implantación y mantenimiento de aplicaciones web de alta calidad (Santamarina, 2011).

2.2.1.1 Ventajas de usar las aplicaciones web.

Las aplicaciones web desempeñan un papel muy importante en el desarrollo de las organizaciones y el control de sus operaciones, debido a que la obtención de la

información de forma inmediata y eficaz contribuye al éxito y al cumplimiento del alcance de los objetivos dentro de la empresa. A continuación se describen las ventajas que representa el uso de las aplicaciones web según la compañía de marketing digital en línea (eNubes Comunicación, S.L, 2013):

- Reducción de costos, ya que no hace falta tener ordenadores muy potentes ni la compra de un software.
- Ahorro de tiempo, se pueden realizar las tareas sin necesidad de descargar ni instalar ningún programa.
- Evita los problemas de compatibilidad entre sistemas, solo hace falta tener un navegador web actualizado para poder utilizarlas.
- No ocupan espacio en nuestro disco duro porque están alojadas en la Nube.
- Están siempre actualizadas, cada vez que nos conectamos accedemos siempre a la versión actualizada automáticamente.
- No consume nuestros recursos, al no estar alojada en nuestro equipo, las operaciones que hace la aplicación no consumen nuestros recursos.
- Multiplataforma, cualquier sistema operativo es válido. Se pueden usar desde cualquier sistema operativo porque sólo es necesario tener un navegador.
- Portabilidad, es independiente del ordenador donde se utilice (un PC de sobremesa, un portátil) porque se accede a través de una página web (sólo es necesario disponer de acceso a Internet). La reciente tendencia del acceso a las aplicaciones web a través de teléfonos móviles requiere sin embargo un diseño específico de los ficheros CSS para no dificultar el acceso de estos usuarios.

- La disponibilidad suele ser alta porque el servicio se ofrece desde múltiples localizaciones para asegurar la continuidad del mismo.
- Seguridad frente a virus y hackers, Los virus no dañan los datos porque éstos están guardados en el servidor de la aplicación.
- Colaboración, gracias a que el acceso al servicio se realiza desde una única ubicación y es sencillo, permite la compartición de datos por parte de varios usuarios, tiene mucho sentido, por ejemplo, en aplicaciones online de calendarios u oficina.
- Permiten compartir información con otros usuarios.
- Son fáciles de usar gracias a las herramientas de formación online.
- Constituyen un nuevo canal de publicidad.
- Si son de pago, se pueden realizar pagos online con tarjeta de crédito, PayPal, entre otros.
- Ofrecen información sobre el comportamiento del usuario que se puede utilizar para las estrategias de marketing y publicidad.

2.2.1.2 Lenguajes de diseño y programación web.

Actualmente existen diferentes lenguajes de programación para desarrollar en la web, estos han surgido debido a las tendencias y necesidades de las plataformas. Desde los inicios de Internet, surgieron diferentes demandas por los usuarios y se dieron soluciones mediante lenguajes estáticos, a medida que pasó el tiempo, las tecnologías fueron desarrollándose y surgieron nuevos problemas a dar solución, esto dio lugar a desarrollar lenguajes de programación para las web dinámicas, que permitieran interactuar con los

usuarios y utilizarán sistemas de Bases de Datos. A continuación se presenta una introducción a los diferentes lenguajes de programación para la web.

HTML5, según la revista Bitelia (2013) es un lenguaje usado para estructurar y presentar el contenido para la web, es uno de los aspectos fundamentales para el funcionamiento de los sitios, pero no es el primero, es la quinta revisión del estándar que fue creado en 1990, a fines del año pasado, la W3C lo recomendó para transformarse en el estándar a ser usado en el desarrollo de proyectos venideros. HTML5 está relacionado también con la entrada en decadencia del viejo estándar HTML 4, que se combinaba con otros lenguajes para producir los sitios que podemos ver hoy en día. Con HTML5, tenemos otras posibilidades para emplear y sin utilizar tantos recursos. Con HTML5, también entra en desuso el formato XHTML, dado que ya no será necesaria su implementación. HTML5 es un nuevo concepto para la construcción de sitios web y aplicaciones en una era que combina dispositivos móviles, computación en la nube y trabajos en red.

Ventajas de usar html5 (NortiPC, 2013):

- Al ser el código más sencillo y simplificado, cargan más rápido las páginas en el navegador.
- Las páginas y los elementos que contienen, se ven perfectamente en todos los navegadores. La gran mayoría de los navegadores de los teléfonos Smartphone y las tabletas, son compatibles con HTML5.
- Los plugins, widgets y botones que ofrecen los desarrolladores de las redes como Facebook, Twitter y otras, escritos en HTML 5 funcionan excelentemente, con más opciones que los clásicos en XHTML o que los iframes.

- Es posible insertar directamente videos en las páginas sin tener que acudir a los iframes o usar la etiqueta object.
- HTML 5 incluye etiquetas orientadas principalmente a los buscadores, para facilitarles comprender el contenido de las páginas, lo que beneficia, por ejemplo: header, footer, article, nav, entre otros.
- Permite la Geolocalización del usuario.

CSS3. El consorcio internacional (W3C, 2013) define a CSS como un lenguaje para definir el estilo o la apariencia de las páginas web escritas con HTML o de los documentos XML, donde el contenido se encuentra separado de la presentación. Los Estilos definen la forma de mostrar los elementos, CSS permite a los desarrolladores Web controlar el estilo y el formato de múltiples páginas Web al mismo tiempo. Cualquier cambio en el estilo marcado para un elemento en la CSS afectará a todas las páginas vinculadas a esa CSS en las que aparezca ese elemento.

Actualmente, CSS3 (hojas de estilo en cascada nivel 3) ofrece una gran variedad de opciones muy importantes para las necesidades del diseño web actual, desde opciones de sombreado y redondeado, hasta funciones avanzadas de movimiento y transformación.

JavaScript (Libros del web, 2013) es un lenguaje de programación que se utiliza principalmente para crear páginas web dinámicas. Una página web dinámica es aquella que incorpora efectos como texto que aparece y desaparece, animaciones, acciones que se activan al pulsar botones, ventanas con mensajes de aviso al usuario, entre otros

eventos. Técnicamente, JavaScript es un lenguaje de programación interpretado, por lo que no es necesario compilar los programas para ejecutarlos, en otras palabras, los programas escritos con JavaScript se pueden probar directamente en cualquier navegador sin necesidad de procesos intermedios, su ejecución dependerá de la capacidad que tenga el navegador para interpretar el código JavaScript.

Existen varias razones para utilizar JavaScript dentro de los documentos HTML según Bodadilla (2000) , nos permite agregar funcionalidades a las páginas que de otra forma sería difícil de realizar (y en ocasiones imposible). En algunos casos se puede utilizar CGI (en especial cuando nos comunicamos con el servidor) en lugar de JavaScript, pero CGI es más difícil de aprender y utilizar. Como otros lenguajes de script que aumentan las capacidades de las aplicaciones con las que trabajan, JavaScript extiende las páginas html más allá de su uso normal. Nos permite dinamizar los sitios web, gracias a su gran flexibilidad. En un sitio web estándar podemos obtener más información al pulsar sobre un enlace hipertexto, que indica al servidor el envío hacia otra página. En una página más interactiva, podemos rellenar un formulario, enviar los resultados al servidor y esperar una respuesta. En cualquier caso, debemos esperar una nueva página desde el servidor. JavaScript puede proporcionar información instantánea, si necesidad de esperar por una respuesta del servidor.

Capítulo III. Metodología

3. Metodología

La documentación del sistema SIGAF se fundamenta sobre una metodología que permite plasmar cada una de las actividades elaboradas durante el desarrollo del sistema, se analizaron diferentes metodologías y se llegó a la conclusión de que Métrica 3 era una opción viable para documentar las fases de desarrollo, puesto que los requerimientos que esta metodología establece se empanan con los del proyecto, dichos requerimientos son la conformación de un equipo de trabajo, la involucración de los usuarios finales en el desarrollo, la definición de roles y responsabilidades de los involucrados con el fin de trabajar de una manera organizada y un punto importante a favor de Métrica 3, es el alcance que se tiene sobre la documentación de la metodología.

Métrica Versión 3, proporciona la estructura para documentar el proceso de desarrollo de sistemas de información, cada una de las etapas de desarrollo se documentan por separado, a medida que se avanza en los módulos del sistema SIGAF se añade la información correspondiente a cada documento.

De los tres procesos principales de esta metodología, se tomaron dos de ellos, la planificación de sistemas de información donde se estableció la metodología para la organización del desarrollo del proyecto y el proceso de desarrollo de sistemas de información donde se definen las tareas que se llevarán a cabo para elaborar cada una de las fases del desarrollo de SIGAF.

A continuación se describen los procesos de la estructura principal de Métrica 3 apegados al proyecto SIGAF.

3.1 Planificación de Sistemas de Información.

Se analizaron las diferentes metodologías que conllevan a desarrollar productos de software de calidad, para determinar que metodología es la más apropiada en la administración del desarrollo del Sistema de Gestión Académica FCA, se tomó como referencia las características del proyecto, tales como su magnitud, la cantidad de personas involucradas, la necesidad de realizar reuniones frecuentemente para analizar la información recabada y retroalimentar, revisar pendientes en cada reunión, entre otros aspectos, al considerar estos elementos se llegó a la conclusión de que SCRUM era la metodología adecuada.

El empleo de la metodología SCRUM en el sistema SIGAF aporta la ventaja de que se pueden gestionar las expectativas del usuario a medida que se avanza en el proyecto, este puede comprobar si se está cumpliendo con sus expectativas y retroalimentar para que se realicen las adecuaciones necesarias que conlleven a lograr el objetivo planteado, lo cual mitiga los riesgos de que no se cumpla con lo que el usuario espera, es una metodología flexible y adaptiva, hay una mayor productividad en los miembros del equipo de trabajo ya que al sincronizar el trabajo diariamente se resuelven problemas que pudieran impedir cumplir con los objetivos de las iteraciones realizadas. Como se establece en la metodología SCRUM, se definieron roles y responsabilidades dentro del proyecto, el primer rol lo toma el líder del proyecto quien representa a las personas interesadas en los resultados de este, colabora con el equipo para planificar,

revisar y dar detalle a los objetivos de cada iteración, los demás roles están asignados al equipo de trabajo, estos son el analista, diseñador de interfaz gráfica de usuario, diseñador de base de datos, programador, pruebas e implementación y documentación.

Al inicio del proyecto se establecieron las iteraciones, las fechas y actividades para llevar acabo cada una de estas, así como la fecha de los entregables, todas estas especificaciones quedaron asentadas en un documento puesto al alcance de los involucrados. Para dar seguimiento a dichas iteraciones, se realizan reuniones cortas 2 o 3 veces por semana para revisar el avance de la iteración en curso y según sea el caso pasar a la siguiente, en estas se elabora una minuta que explica el objetivo, lugar, fecha, duración, personas que asistieron y los avances, cambios y pendientes que se suscitaron durante la reunión.

Durante el desarrollo de cada iteración o sprint se involucra la ejecución de las actividades y tareas de cada uno de los roles; en lo que respecta a documentación, al final del sprint queda documentado el módulo desarrollado y se realiza la parte del manual de usuario y técnico correspondiente a este.

El proceso de desarrollo de la metodología SCRUM inicia por la planeación de los Sprint en base a los requerimientos del cliente, se le debe dar seguimiento a los sprint diariamente y cuando finalicen debe producirse un incremento al producto, es decir, se le agregan más funcionalidades al software en la medida que los sprint se desarrollan.

3.2 Desarrollo de Sistemas de Información

El proceso de desarrollo de sistemas de información está compuesto por el análisis, diseño, construcción, implantación y aceptación. La documentación de cada una de estas etapas quedó definida en la planeación del proyecto donde se identificó el contenido de cada uno de los entregables y se establecieron las fechas para su entrega, cada una de las etapas mencionadas se documenta por separado y contienen la información correspondiente a la etapa por cada uno de los módulos del sistema.

A continuación se enlistan los Sprint mediante los cuales se desarrolló el sistema y la documentación del proyecto SIGAF:

Sprint 1. Creación de la interfaz para la consulta del manual de usuario.

Para desarrollar la interfaz de ayuda se realizaron las siguientes actividades de diseño:

- La elaboración del boceto de la interfaz mediante la herramienta Balsamiq.
- Maquetación del diseño, esta actividad consiste en la distribución de los elementos de cada una de las páginas web para lo que se empleará HTML5 y CSS3 así como JavaScript para darle funcionalidad a los controles.

Sprint 2. Plan de estudios.

Documentar las fases de desarrollo mencionadas a continuación conforme se realiza el sprint:

- Análisis.
- Diseño de la interfaz de usuario.

- Diseño de base de datos.
- Programación.
- Pruebas.
- Elaboración del manual técnico y de usuario correspondiente al sprint.
- Inserción del contenido en la interfaz de consulta del manual de usuario.

Sprint 3. Carga académica.

Sprint 4. Disponibilidad docente

Sprint 5. Login y Usuarios.

Del Sprint dos al cinco, se documentan cada uno de los puntos mencionados en el Sprint dos conforme se desarrollan.

En lo que respecta al manual de usuario, la presentación de información de ayuda para los usuarios está situada en cada uno de los módulos del sistema, en cualquier parte de la interfaz gráfica que se encuentre el usuario podrá pulsar la opción de ayuda y le aparecerá una ventana modal con una lista de las funcionalidades correspondientes a la opción seleccionada. La herramienta empleada para la creación de ayuda es Sublime mediante el uso de Html, CSS3 y JavaScript.

Capítulo IV. Desarrollo

4. Desarrollo

4.1 Creación de la interfaz gráfica para la consulta del manual de usuario

4.1.1 Boceto para la sección de ayuda

Como idea inicial, se desarrolla un boceto donde se define la estructura que tendrá la interfaz, en la figura 2 se muestra el boceto de la pantalla principal de la ayuda el cual se desarrolló en la herramienta Balsamiq, en este se le presenta al usuario la opción de elegir el módulo que desea consultar por medio de cuadros identificados por el nombre y una imagen representativa, es la misma que se usa en el menú principal cuando el usuario inicia sesión.

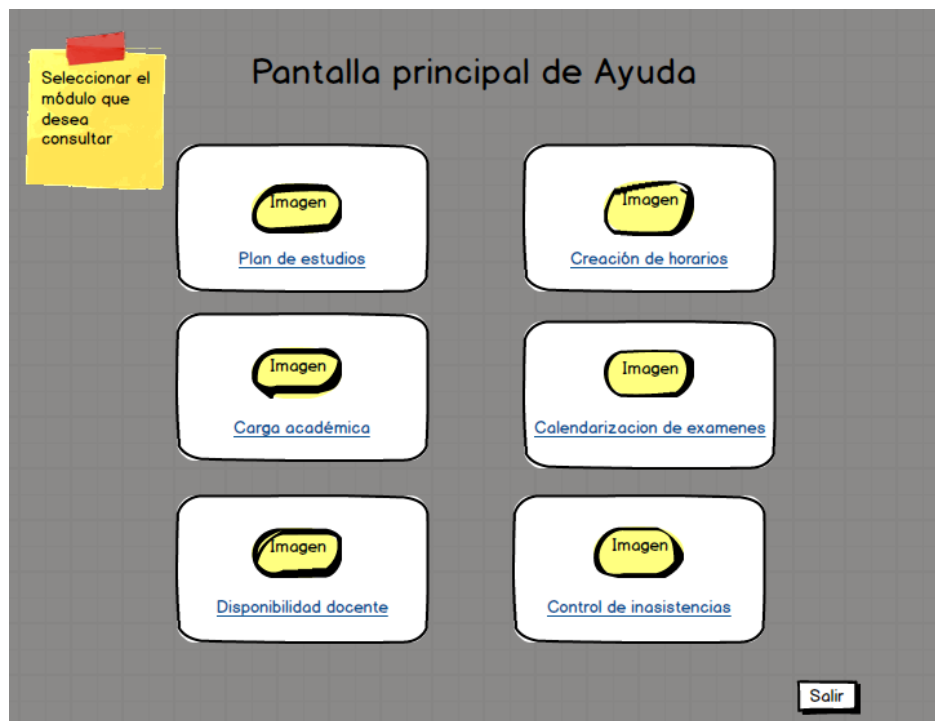


Figura 2. Pantalla principal de ayuda.

La figura 3 muestra el boceto con las funcionalidades, preguntas frecuentes y el glosario del módulo que el usuario haya seleccionado.

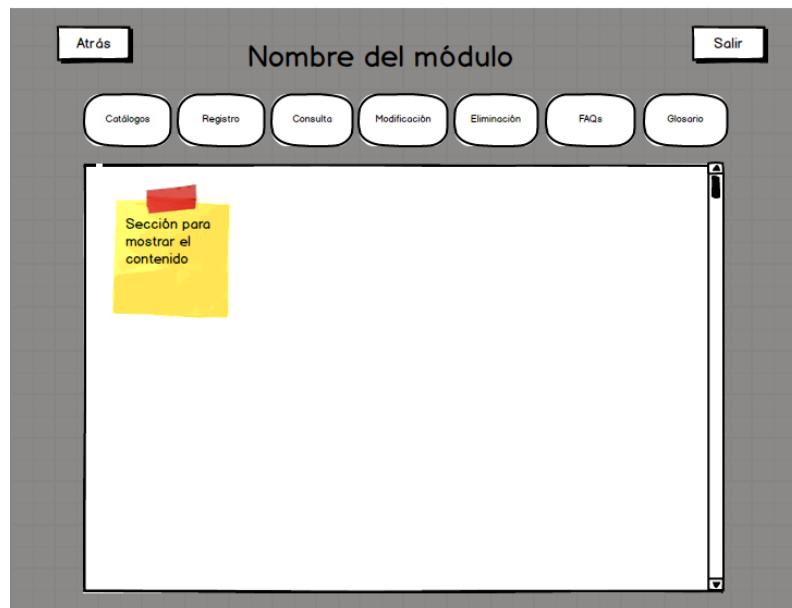


Figura 3. Pantalla con las opciones principales de ayuda.

4.1.2 Ubicación de la ayuda

Dentro de la pantalla principal del sistema hay una sección para consultar el manual de usuario como se aprecia en la figura 4.



Figura 4. Pantalla principal del Sistema.

Otra forma de ubicar la ayuda, dentro de cada uno de los módulos hay una opción identificada con el nombre de manual, esta opción direcciona al usuario a la ayuda del módulo correspondiente.

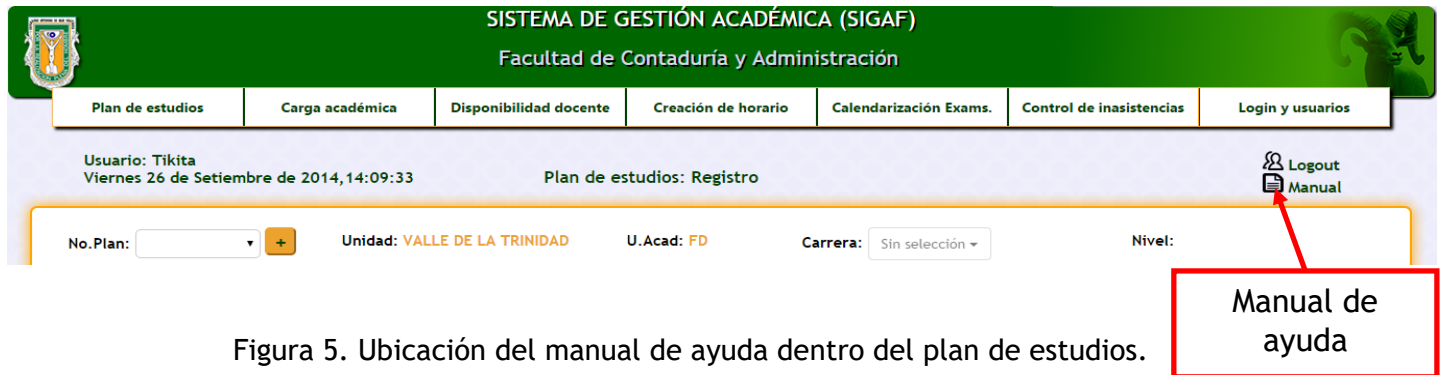


Figura 5. Ubicación del manual de ayuda dentro del plan de estudios.

4.1.3 Pantalla inicial de la ayuda

La pantalla inicial de la ayuda contiene las opciones de los módulos que forman al sistema (lo cual podemos observar en la figura 6), dichas opciones direccionan al usuario a la ventana correspondiente según la opción seleccionada.



Figura 6. Pantalla inicial de la ayuda.

4.1.3.1 Creación de la pantalla inicial.

La pantalla inicial está compuesta por dos divs principales, el primero llamado “modal” se emplea para formar la ventana modal que da el efecto de oscurecimiento y el otro llamado “ventanacompleta” para la inserción de los elementos de esta ventana como lo son cada una de las opciones que se le presentan al usuario, los títulos, el botón de salir y el estilo en general.

Para el contenido se creó un div llamado “contenedor”, este div encierra las opciones para consultar los módulos, cada una de estas opciones está compuesta por un div identificado por el nombre del módulo al que hacen referencia y el link hacia la ventana que direccionarán.

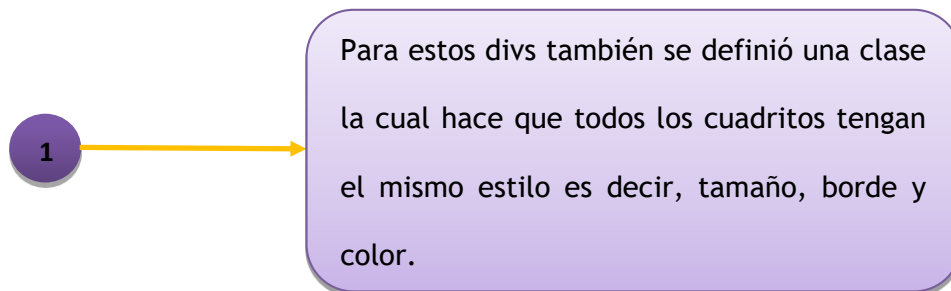
Para crear el contenido y darle estilo se emplean dos elementos, los id y las clases; los id se emplean para identificar al elemento y dar estilo exclusivamente a este y las clases se utilizan para dar estilo a diferentes elementos que tengan asociada dicha clase.

A continuación se muestra una tabla para identificar los elementos que componen la pantalla inicial de la ayuda:

Tabla 1. Elementos que componen la pantalla inicial de la ayuda.

Elemento	Nombre	Descripción	ID o Clase
div	modal	Se emplea para crear y darle estilo a la ventana modal.	Id
div	ventanacompleta	Representa a los elementos de la ventana completa de la pantalla inicial de ayuda.	Id
div	salir	Elemento empleado para ubicar la imagen con el icono de la X la cual envía al usuario a la pantalla principal del sistema.	Id
div	linea1 y línea2	Representas las líneas de color amarillo y verde que dan estilo al encabezado del manual de usuario.	Id
div	contenedor	Contiene las opciones de los módulos.	id
div	plan	Elemento empleado para crear la opción de plan de estudios, mediante este se inserta la imagen, la etiqueta de “Plan de estudios” y el enlace al cual enviará al usuario.	Id y clase
div	carga	Elemento empleado para crear la opción de carga académica, mediante este se inserta la imagen, la etiqueta de “Carga académica” y el enlace al cual enviará al usuario.	Id y clase
div	disponibilidad	Elemento empleado para crear la opción de disponibilidad del docente, mediante este se inserta la imagen, la etiqueta de “Disponibilidad docente” y el enlace al cual enviará al usuario.	Id y clase
div	horarios	Elemento empleado para crear la opción de creación de horarios, mediante este se inserta la imagen, la etiqueta de “Creación de horarios” y el enlace al cual enviará al usuario.	Id y clase
div	calendarización	Elementos empleado para crear la opción de calendarización de exámenes, mediante este se	Id y clase

		inserta la imagen, la etiqueta de “Calend.de exams.” y el enlace al cual enviará al usuario.	
div	inasistencias	Elementos empleado para crear la opción de control de inasistencias, mediante este se inserta la imagen, la etiqueta de “Control de inasist.” y el enlace al cual enviará al usuario.	Id y clase
div	pie	El div pie se emplea para insertar y dar estilo al texto ubicado en el pie de página.	Id
div	overlay	Este div es el que da el efecto de oscurecimiento a la ventana modal.	clase
h1		Elemento que da estilo al encabezado principal.	
h2		Elemento utilizado para dar estilo a los nombres de los módulos.	
span		Empleado para darle estilo al texto asociado al encabezado principal y al pie de página.	



El orden de los elementos de la tabla 1 está en función de la estructura de la maquetación. El archivo de la pantalla inicial se denomina principal.html y tiene asociado una hoja de estilo llamada inicio.css.

4.1.4 Pantalla para cada uno de los módulos.

La siguiente figura muestra la ventana de ayuda correspondiente al módulo de plan de estudios.



Figura 7. Ayuda del módulo plan de estudios.

La sección de catálogos muestra el registro, modificación y eliminación, para el resto de las opciones el nombre hace alusión al contenido. Las opciones se manejan por medio de hipervínculos para que el usuario tenga más facilidad de desplazamiento.

La figura anterior muestra la interfaz general del manual de usuario para cada uno de los subsistemas.

Opciones de la interfaz de usuario:

1. Catálogos. Los subsistemas contienen catálogos los cuales alimentan al sistema con los datos necesarios para lograr la gestión de cada uno de los módulos, desde el registro hasta la salida de impresión de la información. Esta sección comprende los pasos y el orden que se debe seguir para dar de alta correctamente cada uno de los catálogos.
2. Registro. El registro está compuesto por la inserción de los datos en los campos solicitados con sus respectivas restricciones y validaciones, esta opción muestra los pasos necesarios para llevar esta funcionalidad.
3. Consulta. Le presenta al usuario los criterios mediante los cuales se puede realizar una búsqueda y los pasos para realizar dicha acción.
4. Modificación. Contiene los pasos para indicarle al usuario de qué manera puede realizar la actualización de los datos y bajo que restricciones.
5. Eliminación. Existen dos tipos de eliminación, física y lógica, esta opción muestra el seguimiento de los pasos mediante los cuales se puede realizar una eliminación.
6. Preguntas frecuentes (FAQ). Esta sección contiene las preguntas frecuentes para cada uno de los subsistemas con sus respectivas respuestas.
7. Glosario. Contiene la definición de palabras técnicas necesarias para facilitar la comprensión del usuario y la terminología asociada al módulo correspondiente.

Las definiciones del glosario se encuentran ordenadas alfabéticamente.

4.1.4.1 Creación de la plantilla para cada uno de los módulos.

Cada uno de los módulos que componen el sistema tiene una ayuda asociada, dentro de esta se encuentran los pasos a seguir para realizar cada una de las funcionalidades del módulo, en la figura 8 se muestran las opciones de la interfaz, la sección de catálogos contiene los pasos para la generación de los catálogos que alimentarán al sistema, las siguientes opciones corresponden a las funcionalidades de registro, consulta, modificación y eliminación de información, está también la opción de preguntas frecuentes y por último un glosario donde se especifican los términos que el usuario necesita conocer para el correcto funcionamiento de la interfaz y manejo de información.

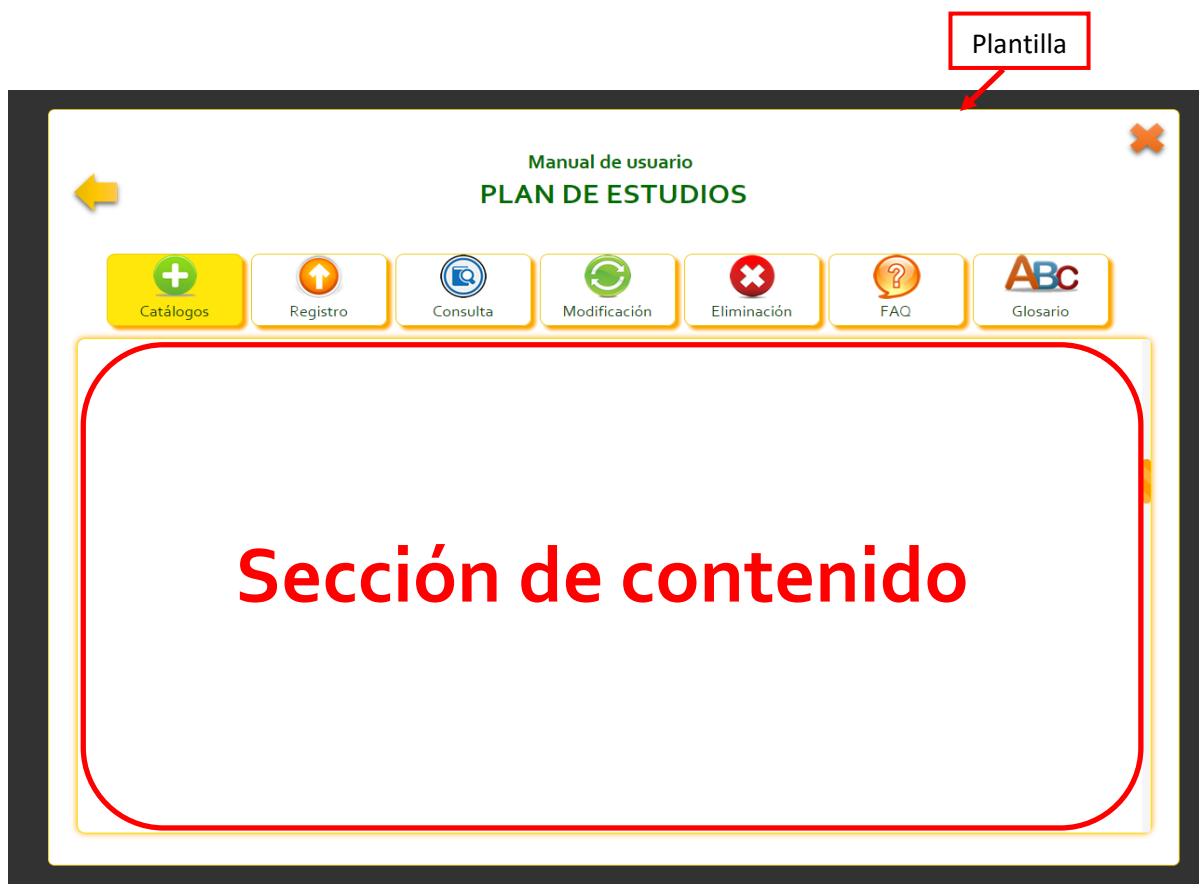


Figure 8. Plantilla para cada uno de los módulos.

La estructura general de esta ventana está compuesta por los siguientes elementos:

```
<div id=container>
  <section class="tabs">
    <input id="tab-1"
  /><label>Catálogos</label>
    <input id="tab-2" /><label>Registro</label>
    ....
    <div class="content">
      <div class="content-1"> Catálogos
      <!--Contenido de los catálogos-->
      </div>
      <div class="content-2"> Registro
      <!--Contenido del registro-->
      </div>
      ....
    </div>
  </section>
</div>
```

Figura 9. Estructura general de la plantilla de ayuda para cada uno de los módulos.

El color azul denota a los elementos que conforman la estructura de la ventana, el color verde hace referencia al nombre del elemento.

Cada uno de los input que se muestran son las opciones (catálogos, registro, consulta, modificación, etc.) que se le presentan al usuario, en caso de ser necesario añadir más opciones solo se tienen que agregar más input y ajustar el tamaño de la sección “tabs”.

Las opciones de contenido deben corresponder a las que se le presentan al usuario; el contenido está formado por:

Tabla 2. Elementos que integran el contenido de la plantilla.

Elemento	Descripción
Texto	Identificado por la etiqueta <p>.
Listas	Se definieron listas con número y con viñeta <ul class="lista-num"> <ul class="lista-vineta"> .
br	Salto de línea

Se definió un archivo para cada uno de los módulos los cuales se pueden ubicar con las iniciales de estos (plan estudio es pe.html, carga académica es ca.html y así sucesivamente), cada uno de estos archivos tiene asociado dos hojas de estilo, demotab.css contiene el estilo de la ventana modal y estilotab.css es el que da el estilo general a la plantilla y al contenido.

Para emplear un estilo es necesario que en el archivo head del html se encuentre la referencia de dicho estilo, por ejemplo:

```
<!DOCTYPE html >
<head>
    <link rel="stylesheet" href="css/demotab.css" type="text/css" media="screen" />
    <link rel="stylesheet" href="css/estilotab.css" type="text/css" media="screen" />
</head>
<body>
</body>
</Html>
```

Figure 10. Referencia de las hojas de estilo.

4.2 Documentación del Sistema

Para el sistema SIGAF se emplean dos procesos de los tres que componen a métrica 3, la planificación de sistemas de información y el desarrollo de sistemas de información, la etapa de mantenimiento no es aplicable para la fase inicial de SIGAF, será en la segunda versión donde se le de mantenimiento al sistema.

En la fase de planificación se estableció la organización del proyecto, en esta se definen los roles y la planificación de las actividades para cada uno de ellos, se establecieron tiempos de entrega y se contemplaron los recursos necesarios para llevar a cabo el proyecto en condiciones óptimas.

En cuanto a la documentación del desarrollo del sistema, Métrica Versión 3 define las actividades que se deben de seguir para el desarrollo del proceso de software de acuerdo a las fases de análisis, diseño, construcción e implantación, estas se tomaron como base para establecer el contenido de la documentación en cada una de las fases de desarrollo del proyecto SIGAF.

Métrica Versión 3 tiene un enfoque orientado al proceso y se encuentra enmarcada dentro de la norma ISO 12207 la cual se centra en la clasificación y definición de los procesos del ciclo de vida del software; esta metodología descompone cada uno de los procesos en actividades, y éstas a su vez en tareas; para cada tarea se describe su contenido haciendo referencia a sus principales acciones, productos, técnicas, prácticas y participantes.

El orden asignado a las actividades no se contempla como secuencia en su realización, ya que éstas pueden realizarse en orden diferente a su numeración o bien en paralelo,

sin embargo, no se dará por terminado un proceso hasta no haber finalizado todas las actividades del mismo determinadas al inicio del proyecto.

En cada uno de los documentos se especifica el estándar o la herramienta empleada, posteriormente se define el contenido que involucra la fase a documentar, dicho contenido se construyó bajo un formato estándar que contempla el tipo de letra “Corbel”, el tamaño de letra para el contenido es doce y los títulos trece, el nombramiento de tablas y figuras, entre otros aspectos que permiten desarrollar un trabajo unificado y fácil de entender para el usuario.

Como parte de las actividades del rol de documentación es importante mencionar que para establecer la comunicación y gestionar los cambios realizados dentro del proyecto se empleó la herramienta Base Camp (mostrada en la figura 11), esta es un gestor de tareas para equipos que trabajan de forma colaborativa, en esta plataforma se asignan actividades a los roles con sus respectivas fechas lo cual permite llevar un control de las actividades realizadas, también se puede obtener reportes de rendimiento de cada uno de los involucrados, cuenta con una sección para subir los archivos clasificados por fases; cada uno de los roles puede hacer aportaciones o comentar sus dudas, cada vez que alguien comenta le llega un correo a todos los involucrados.



Figura 11. Pantalla principal de Base Camp.

En la siguiente figura se aprecia la lista de las personas participantes en el proyecto con sus respectivos datos tales como nombre, teléfono y correo electrónico proporcionados para entablar una mejor comunicación.

Sistema de Gestión Académica FCA (SIGAF)

Teléfonos y contactos de equipo de trabajo SIGAF

Delia Yadhira Gambino Canales - Diseño Base de Datos
Teléfono : 6340236 , 9745196
Correo : deliaygambino@hotmail.com , delia.gambino@uabc.edu.mx
Emma Sofía Castillejos Caballero - Análisis
Teléfono: 6641889221 , 6865679
Correo: emma.castillejos@uabc.edu.mx , emma721014@hotmail.com
Beatriz Ramirez Moreno - Lider de proyecto
Teléfono: 6642941453
Correo: ramirezmb@hotmail.com , ramirezmb@uabc.edu.mx
Cynthia Betsabé Duarte Frausto - Diseño Frontend
Teléfonos: 6641878561 , 6233987
Correo: zyntyaduarte@hotmail.com
Unice Espinoza Bastidas - Documentación
Teléfono: 6643640825
Correo: unice19@hotmail.com
Edgar Iván Avila Garrido - Diseño Backend
Teléfono: 6450706
Correo: wolfogan@gmail.com , iavila85@uabc.edu.mx , logan_602@hotmail.com

Want to make changes?

[Edit this document](#)

Unice Espinoza was the last person to edit this document on Jan 26, 2014 ([See what changed](#)).

Version history

- Unice E. Jan 26 at 9:31pm
- Unice E. Jan 7 at 1:12pm
- Edgar I. Dec 19 at 10:28pm
- Edgar I. Dec 19 at 1:19pm

[Delete...](#)

Figura 12. Lista de los integrantes del proyecto participantes en Base Camp.

4.2.1 Control de cambios

A medida que se avanza en el desarrollo del proyecto se presentan cambios, los cuales deben quedar asentados para su oportuno seguimiento, dicho control de cambios se administra mediante el uso de minutas que se elaboran durante las reuniones de trabajo con el objetivo de analizar los avances del proyecto y determinar si hay necesidad o es viable realizar algún cambio. Desde que empezó a desarrollarse el sistema han surgido diversos cambios, desde la presentación y aprobación de la fase de análisis, la revisión de las diferentes propuestas presentadas por parte de la fase de diseño de la interfaz gráfica del sistema, el empate que se realizó entre la base de datos y la interfaz gráfica de usuario, entre otros aspectos que han sido de gran importancia para el avance de este proyecto. En la figura 13 se muestra el contenido de las minutas.

UNIVERSIDAD AUTÓNOMA DE BAJA CALIFORNIA
FACULTAD DE CONTADURÍA Y ADMINISTRACIÓN
MAESTRÍA EN TECNOLOGÍAS DE LA INFORMACIÓN Y LA COMUNICACIÓN

Sistema de Gestión Académica FCA

Tijuana Baja California, 19 de Junio del 2014

RESUMEN DE REUNIÓN

Objetivo de la reunión	Revisión de avances programados del módulo de Plan de Estudios.
Lugar, fecha y hora de reunión	Facultad de Contaduría y Administración (FCA), sala de juntas, jueves 19 de junio del 2014, 04:30 – 5:30 hrs.

ASISTENTES

Nombre	Rol	Firma

Resumen de la reunión:

AVANCES

ACUERDOS

PENDIENTES

Figura 13. Contenido de las minutas.

El uso de las minutas no solo permite llevar un control de los cambios, si no que mantiene informado a todo el equipo involucrado en el proyecto sobre los temas que se trataron en las reuniones, los avances obtenidos, los acuerdos en los cuales se incluyen los cambios acordados y las actividades pendientes por hacer.

4. 2. 2 Plan de estudios

Es el conjunto de actividades de enseñanza-aprendizaje que comprende la descripción general de los contenidos de un programa de unidad de aprendizaje (conjunto de materias del mapa curricular).

4.2.2.1 Fase de análisis

La documentación de análisis se realizó bajo el estándar IEEE 830-1998 el cual se enfoca en la especificación de requerimientos de software, el estándar en si propone la estructura del contenido para documentar los requerimientos, sin embargo, se realizaron algunas adaptaciones a este estándar combinándolo con las actividades que establece la metodología Métrica Versión 3 en la que se define la estructura para documentar todo el desarrollo de un sistema y la cual se tomó como referencia para documentar SIGAF.

Para realizar la representación gráfica del sistema se empleó la herramienta Lenguaje de Modelado Unificado (UML); en lo que respecta a la fase de análisis se utilizaron los diagramas de casos de uso, clases y objetos.

La representación de procesos se realizó mediante la herramienta “Notación para el Modelado de Procesos de Negocio (BPMN por sus siglas en inglés)” la cual hace referencia a un estándar internacional donde se emplea notación grafica para describir la lógica de los pasos de un proceso de negocio.

La figura 14 muestra el contenido de la fase de análisis, dicho contenido se emplea para cada uno de los subsistemas.

Análisis del Sistema de Información (ASI)

Especificación de estándares y normas

Contenido

1. Definición del subsistema

- 1.1 Determinación del alcance del subsistema
- 1.2 Definiciones, acrónimos y abreviaturas
- 1.3 Identificación y características de los usuarios finales

2. Establecimiento de requisitos

- 2.1 Requisitos específicos

3. Análisis de casos de uso

- 3.1 Especificación de casos de uso
- 3.2 Descripción de la interacción de objetos

4. Análisis de clases

- 4.1 Identificación de clases, atributos y sus relaciones

5. Elaboración del modelo de procesos

- 5.1 Obtención del modelo de procesos del sistema

6. Anexos

Figura 14. Contenido de la fase de análisis.

El contenido de análisis inicia con la definición del sistema donde se describe el alcance de este, las definiciones, acrónimos y abreviaturas las cuales sirven para ayudar al usuario a interpretar de mejor manera el contenido, la identificación de los usuarios y sus características con el fin de conocer a fondo las necesidades de quienes harán uso del sistema. En el establecimiento de requisitos se especifican los requerimientos funcionales y no funcionales del subsistema ordenados en forma de tablas, cada una de estas se identifica por la palabra “tabla” seguida de las iniciales del módulo y un número

diferenciador, la nomenclatura para identificar cada requerimiento está compuesta por las iniciales del requerimiento funcional (RF) o no funcional(RNF) seguido de las iniciales del módulo y un número identificador del requerimiento, el resto del contenido está compuesto por el nombre del requerimiento, las características, descripción, los requerimientos no funcionales a los cuales está asociado y el nivel de prioridad que este tiene.

En el análisis de casos de uso se identifican cada uno de los casos de uso del subsistema, representados por medio de diagramas elaborados en UML así como la identificación e interacción que hay entre los objetos. En el análisis de clases se identifican cada una de las clases del subsistema con sus respectivos atributos y relaciones, y se realiza un modelo de procesos del subsistema que permite conocer de manera general como se lleva a cabo el proceso de cada uno de los módulos que componen el sistema, por último se encuentran los anexos los cuales se tomaron como referencia para fundamentar y complementar la fase de análisis.

Antes de empezar a desarrollar el diseño de la interfaz gráfica de usuario se verificaron los requerimientos de cada uno de los subsistemas con el usuario final los cuales quedaron aprobados y firmados, esto queda asentado en la carta de aceptación de cada uno de los subsistemas.

Para consultar la documentación de análisis, vaya a la sección de resultados y abra el documento “Análisis del sistema SIGAF”.

4.2.2.2 Fase de diseño del sistema (Interfaz Gráfica de Usuario y BD)

Al igual que en la fase de análisis, en el diseño de la GUI y de la base de datos también se especifican los estándares y reglas de diseño empleadas, en lo que respecta a la GUI, primeramente se realiza el diseño lógico de la interfaz para el cual se utilizó la herramienta UML, se desarrollaron los diagramas de secuencia para representar la interacción que hay entre los objetos del sistema y el diagrama de estados para representar los diferentes estados en los que se procesa un registro de unidad de aprendizaje, estos diagramas sirven como referencia a las fases del diseño de base de datos y programación.

Una vez que se realizó la parte lógica de la interfaz se procedió a realizar los bocetos y el diseño gráfico para posteriormente maquetar el diseño, esta actividad consiste en darle un formato a los documentos, a todo el conjunto de elementos que lo componen, las imágenes, textos; este proceso se llevó a cabo mediante los lenguajes HTML, CSS y JavaScript.

Por otro lado, la base de datos se desarrolló en el lenguaje MySQL, para su elaboración se tomó como referencia el análisis de requerimientos y el modelado lógico de la interfaz de usuario. La base de datos está formada por las tablas, procedimientos y triggers; el nombramiento de las tablas se realizó en base a los datos que se almacenan en estas, los procedimientos se identifican por la nomenclatura SP que hace referencia a un procedimiento almacenado, el nombre de la tabla que afecta y la acción a realizar.

La figura 15 muestra el contenido de la fase de diseño de la GUI así como de la base de datos. El contenido inicia con la definición de la arquitectura del sistema, posteriormente se indican los puntos concernientes al desarrollo de la GUI, desde que se realiza la parte lógica para analizar el comportamiento de cada uno de los subsistemas así como el flujo de la información, hasta la maquetación en la cual quedan definidos los elementos reales que conforman a la interfaz de usuario.

Diseño del Sistema de Información (DSI)

Especificación de estándares y normas de diseño y construcción

- 1. 1. Diseño de la arquitectura del subsistema
 - 1.1.1. Diseño lógico del subsistema
 - 1.1.1.1. Diagramas en UML
 - 1.1.2. Diseño físico del subsistema
 - 1.1.2.1. Mockup
 - 1.1.2.2. Diseño gráfico de la interfaz
 - 1.2.2.3. Maquetación de la interfaz
- 1. 2. Diseño de datos
 - 1.2.1. Diseño conceptual de datos
 - 1.2.1.1. Diccionario de datos
 - 1. 2.2. Diseño lógico de datos
 - 1.2.2.1. Modelo relacional
 - 1.2.2.2. Restricciones de integridad
 - 1. 2.3. Diseño físico de datos
 - 1.2.3.1. Generación de scripts de la BD
 - 1.2.3.2. Generación de triggers
 - 2.2.3. Diseño físico de datos
 - 2.2.3.1 Generación de scripts de la BD
 - 2.2.3.2 Generación de procedimientos

Figura 15. Contenido de la fase de diseño.

Para ver la documentación completa de la fase de diseño vaya a la sección de resultados y abra el archivo “Diseño del sistema SIGAF”.

4.2.2.3 Fase de construcción del sistema (programación)

La documentación de la fase de construcción inicia con la especificación de los estándares y lenguajes de programación empleados para la codificación del sistema; se emplea un marco de trabajo llamado Laravel el cual le permite al programador utilizar una sintaxis elegante y expresiva para crear código de forma sencilla con multitud de funcionalidades. Esta herramienta es una combinación de otros marcos de trabajo con las últimas versiones de PHP. La siguiente figura muestra el contenido de esta fase la cual se aplica para cada uno de los subsistemas que componen SIGAF.

Construcción del Sistema de Información (CSI)

Especificación de estándares y lenguaje de programación

Contenido

1. Preparación del entorno de generación y construcción

- 1.1 Implantación de la Base de Datos Física
- 1.2 Preparación del entorno de construcción

2. Generación del código de los componentes

- 2.1 Generación del código de componentes

3. Ejecución de pruebas unitarias

- 3.1 Preparación del entorno de pruebas unitarias
- 3.2 Realización y evaluación de las pruebas unitarias

Figura 16. Contenido de la fase de construcción (programación).

El contenido contempla la preparación del entorno de generación y construcción en el cual se organiza el ambiente de trabajo al adaptar el código de la interfaz gráfica de usuario y los scripts de la base de datos para empezar a programar, la codificación de cada una de las funcionalidades del sistema y la ejecución de pruebas unitarias en

cuanto a programación para validar que el sistema cumpla con la funcionalidad y no tenga errores.

El control de versiones de la interfaz gráfica de usuario y la programación se lleva mediante la herramienta GitHub, esta permite sincronizar los cambios que se realizan en cada una de las fases de desarrollo empatándolos al generar una nueva versión con las últimas modificaciones realizadas. GitHub, es un servicio de alojamiento de repositorios de software, que lejos de quedarse en esta funcionalidad, ofrece un conjunto de características muy útiles para el trabajo en equipo, se emplea en proyectos de software libre como JQuery, Reddit, Sparkle, Ruby on Rails, Node.js, CakePHP, Redis y otros más.

En cuanto a la documentación del código se utiliza la herramienta php documentor la cual genera los comentarios automáticamente sobre este.

Para ver el contenido de la programación del sistema SIGAF vaya al archivo manual técnico.

4.2.2.4 Fase de pruebas e implantación del sistema

La fase de pruebas e implementación del sistema se encuentra fundamentada por la metodología Métrica V3 en la cual se establecen cada una de las actividades que hay que seguir para esta labor. La figura 17 muestra el contenido de la fase de implantación y pruebas, el cual se aplica para cada uno de los subsistemas de SIGAF.

Contenido

1. Introducción

- 1.1 Propósito
- 1.2 Alcance
- 1.3 Definiciones, Acrónimos, y Abreviaciones
- 1.4 Referencias
- 1.5 Vista Global

2. Planificación de la Entrega

- 2.1 Responsabilidades
- 2.2 Cronograma

3. Recursos

- 3.1 Facilidades
- 3.2 Hardware
- 3.3 Unidad de Implantación
 - 3.3.1 Software de Soporte
 - 3.3.2 Documentación de Soporte
 - 3.3.3 Personal de Soporte

4. Entrenamiento

Figura 17. Contenido de la fase de implantación y pruebas.

El contenido de esta sección está compuesto por el plan de implantación, formación necesaria para la implantación, incorporación del sistema al entorno de operación, carga de datos al entorno de operación, pruebas de aceptación del sistema.

En la sección de plan de implantación se revisa la estrategia de implantación para el sistema, se identifican los sistemas de información que forman parte del objeto de la implantación, para cada sistema se analizan las posibles dependencias con otros proyectos que pudieran condicionar el plan de implantación, se constituye el equipo que realizará dicho proceso y se asignan los recursos humanos necesarios para la instalación, pruebas y aceptación, debe quedar establecido en el plan el rol y responsabilidades de cada una de estas personas.

Una vez constituido el plan, la siguiente actividad es llevarlo a cabo al preparar el ambiente de trabajo necesario para incorporar el sistema en el entorno de operación y cargar los datos necesarios que permitan hacer las pruebas para comprobar que el sistema no tiene errores y funciona de acuerdo a lo establecido en los requerimientos.

A continuación se define los módulos de plan de estudios, carga académica y disponibilidad docente, en estos se contemplan las mismas actividades y características que se emplearon para el subsistema login y usuarios.

4.2.3 Carga académica

Conjunto de unidades de aprendizaje dentro del plan(es) de estudio que se le ofrecen al alumno para cursar en un periodo escolar (semestral).

4.2.4 Disponibilidad docente

Especificación de los datos personales, escolaridad, actividad profesional, capacitación que ha recibido o impartido un docente, así como la especificación de materias y horarios en los que pueden impartir una unidad de aprendizaje.

4.2.5 Login y usuarios

Representa al conjunto de usuarios que se registran en el sistema para que desempeñen determinados roles, de acuerdo a este son las tareas que desarrollan y el acceso que tienen al sistema SIGAF.

4.2.6 Manual de usuario

El manual de usuario comprende cada uno de los pasos que se deben seguir para que el usuario entienda el funcionamiento correcto del sistema SIGAF. El manual de usuario se presenta en formato PDF y por medio de la interfaz del sistema, el contenido del manual inicia con una breve introducción, seguida de la nomenclatura y los conocimientos básicos que debe poseer el usuario para hacer un buen uso del sistema. En la figura 18 se muestra el contenido general del manual.

Contenido

Introducción

Acceso al sistema

1.0 Login y Usuarios

- 1.1 Registro de usuarios
- 1.2 Actualización de usuarios
- 1.3 Consulta de usuarios
- 1.4 Eliminación de usuarios
- 1.5 Cambiar contraseña
- 1.6 Olvidó contraseña

2.0 Catálogos

3.0 Plan de Estudios

- 3.1 Registro de unidades de aprendizaje para el plan de estudio
- 3.2 Actualización de Unidades de Aprendizaje del Plan de Estudio
- 3.3 Consulta de Unidades de Aprendizaje del Plan de Estudio
- 3.4 Eliminación de Unidad de Aprendizaje del Plan de Estudio

Figura 18. Contenido del manual de usuario.

En el contenido del manual de usuario se encuentran los módulos que componen el sistema con sus respectivas funcionalidades. Para acceder al manual de usuario vaya a la sección de resultados y seleccione manual de usuario, otra opción es consultar la interfaz gráfica del sistema.

Capítulo V. Resultados

5. Resultados

Esta tesis está basada en el desarrollo de la documentación del sistema SIGAF, como resultado se obtuvieron los siguientes documentos:

- **Documento de análisis:** el documento de análisis se elaboró bajo el estándar IEEE 830-1998 el cual proporciona las normas para la especificación de requerimientos de software, dicho documento está compuesto por el alcance del sistema, la especificación de los requerimientos funcionales y no funcionales, el modelado de las procesos bajo el estándar UML donde se definen casos de uso y clases que permiten representar el flujo de la información correspondiente a las funcionalidades principales. En este documento se encuentra el análisis de todos los módulos de SIGAF en su versión inicial. El siguiente archivo le permite ver el análisis del sistema con más detalle.



Análisis del sistema
SIGAF.docx

- **Documento de diseño:** el documento de diseño involucra a la interfaz gráfica de usuario y la base de datos, en cuanto a la interfaz gráfica se utilizó UML para el diseño lógico, el cual está compuesto por los diagramas de secuencia y estados, el diseño físico comprende la elaboración de un boceto, el diseño gráfico de la interfaz y la maquetación de esta, con respecto a base de datos la documentación se compone por un diccionario de datos, el modelo relacional, las restricciones de integridad, el diseño físico de datos y la generación de triggers para la bitácora del sistema. Si desea consultar más detalles, puede abrir el siguiente archivo

“Diseño del sistema SIGAF” el cual contiene la documentación completa de cada uno de los módulos.



Diseño del sistema
SIGAF.docx

- **Plan de implantación:** el plan de implantación involucra los requerimientos necesarios de hardware y software para que el sistema se implemente en un ambiente funcional. Si desea más detalle sobre este documento, abra el archivo “Plan de implantación SIGAF”.



Plan de implantacion
SIGAF.docx

- **Manual de usuario:** el manual de usuario contiene los pasos para que el usuario pueda hacer uso correcto del sistema, este se encuentra disponible en formato PDF y también puede ser consultado desde la interfaz del sistema, en el menú principal se le presenta al usuario la opción “manual y ayuda” la cual lo direcciona a elegir el módulo que quiere consultar o en su defecto dentro del mismo módulo hay una sección de ayuda correspondiente a este. Si desea consultar más detalles del manual de usuario abra el siguiente archivo “Manual de usuario SIGAF” o consulte la interfaz del sistema.



Manual de Usuario
SIGAF.docx

- **Manual técnico:** el manual técnico va dirigido a los programadores o las personas que se encargarán de darle mantenimiento al sistema, contiene la relación que

hay entre la interfaz y la codificación que le da vida a esta, el documento inicia con la especificación de estándares y normas de diseño y construcción; el sistema se codificó bajo el entorno Laravel, el control de versiones se llevó acabo con la herramienta GitHub. En el documento se explica cómo esta estructurada la codificación del frontend y backend, si desea ver más detalles abra el siguiente archivo “Manual técnico SIGAF”.



Manual Técnico
SIGAF.docx

Otro de los resultados obtenidos es la interfaz gráfica de usuario para la consulta del manual de usuario en el sistema.

En la figura 19 se observa el menú principal de la sección de ayuda el cual contempla los módulos que componen al sistema.

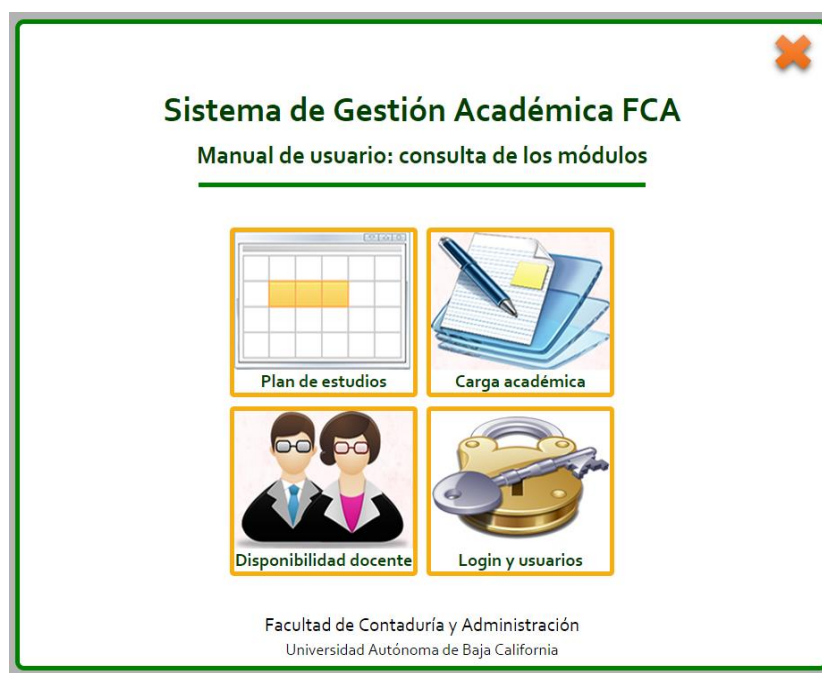


Figura 19. Pantalla principal de la sección de ayuda.

La figura 20 muestra la interfaz para consultar la ayuda del módulo de disponibilidad docente en la opción de preguntas frecuentes. En la figura 21 se observa la interfaz para consultar la ayuda del módulo de carga académica en la opción de modificación.



Figura 20. Interfaz de ayuda del módulo plan de estudios.



Figura 21. Interfaz de ayuda del módulo de carga académica

La figura 22 muestra la interfaz para consultar la ayuda en el módulo de disponibilidad docente en la sección del glosario.



Figura 22. Interfaz de ayuda del módulo disponibilidad docente.

Capítulo VI.

Conclusiones y recomendaciones

6. Conclusiones y recomendaciones

Como resultado del avance del desarrollo del sistema SIGAF y la documentación de este, se concluye que documentar no es una tarea fácil para quien la realiza, sin embargo, el gran trabajo que implica la elaboración de esta se justifica con las ventajas que proporcionan dichos documentos, sobre todo cuando se trabaja sobre un proyecto grande y hay necesidad de establecer acuerdos, elaborar planes de comunicación y generar los documentos que permitan gestionar dicho proyecto. De igual forma resulta interesante para quien requiere conocer con precisión los procesos o actividades que se realizan dentro de una organización, realizar actualizaciones o mantenimientos futuros así como también establecer las bases para posibles auditorias.

Con respecto a la metodología Métrica V3, es una metodología muy completa y al alcance de quien requiera hacer uso de estos documentos para desarrollar un proyecto, esta se basa en estándares de calidad para establecer las actividades que se deben de llevar a cabo en el desarrollo de un sistema de información durante el ciclo de vida de este, dichas actividades se tomaron como referencias para documentar las fases de desarrollo del sistema SIGAF.

La metodología SCRUM se utilizó para el desarrollo del sistema, esta metodología tiene la ventaja de que le permite al usuario final corroborar si el proyecto va encaminado a sus expectativas, se pueden ver resultados a corto plazo, es flexible y adaptativa, permite que el equipo sincronice su trabajo y se ayuden entre sí para resolver problemas que pudieran impedir que se cumple con los objetivos trazados.

La versión inicial del proyecto SIGAF se desarrolló en equipo, es importante mencionar lo que implica un trabajo en conjunto donde se consideran las habilidades y talentos de quienes lo integran, se contemplan más rápido las tareas, por el apoyo que hay entre el equipo, por la gran complejidad del proyecto, porque se pueden aportar nuevas ideas y otras razones que pudieran haber para resaltar lo sustancial que es saber trabajar en equipo, sin embargo, un equipo que no se coordina, no se organiza, por más talento que haya no da soluciones oportunas, como recomendación hay que tener en consideración que trabajar en equipo es muy útil (como se menciona en la metodología SCRUM), pero no es trabajo de unos cuantos, es trabajo de todos y hay que saber adaptarse y aprovecharlo al máximo.

Bibliografía

Alvarez, S. (2006). Desarrollo web. Disponible en <http://www.desarrolloweb.com/articulos/importancia-documentacion.html>, consultado el 12/03/2014.

Bitelia, R. (2013). Entendiendo HTML5: guía para principiantes. Disponible en <http://bitelia.com/2013/05/entendiendo-html5-guia-para-principiantes>, consultado el 12/03/2014.

Bodadilla Jesús, A. A. (2000). HTML Dinámico, ASP y JavaScript a través de ejemplos. Alfaomega.

Braña, F. (2013). Portal programas. Disponible en <http://gratis.portalprogramas.com/KRONOWIN.html>, consultado el 15/03/2014.

Brun, R. E. (1998). Disponible en <http://ibersid.eu/ojs/index.php/scire/article/view/File/1092/1074>, consultado el 23/03/2014.

Caéz, M. F. (2013). Disponible en <http://www.acis.org.co/fileadmin/Conferencias/DocumentacionSoftware.pdf>, consultado el 03/04/2014.

Catalonia Software Solutions. (2013). Disponible en <http://gestionhorarios.catalonia.es/cuadros-horarios.html>, consultado el 15/03/2014.

eNubes Comunicación, S.L. (2013). Disponible en <http://www.enubes.com/desarrollo-de-aplicaciones-web.html>, consultado el 08/04/2014.

Fuentes, M. d. (2011). Disponible en http://web.cua.uam.mx/publicaciones/Notas_Analisis_Requerimiento.pdf, consultado el 15/04/2014.

Grady Booch, J. R. (2006). El Lenguaje Unificado de Modelado UML. Pearson Addison Wesley.

Itequia. (2014). Disponible en <http://www.itequia.com/es/soluciones/alm-software-process/scrum-training/>, consultado el 17/04/2014.

John Freddy Vega, C. V. (2011). Mejorando.la. Disponible en <https://mejorando.la/static/cursos/documentos/mejorandolaweb-guia-html5.pdf>, consultado el 17/04/2014.

Libros del web. (2013). Disponible en http://librosweb.es/javascript/capitulo_1.html, consultado el 18/04/2014.

Manuel Amadeo González Gálvez, L. M. (2013). Disponible en <http://cybertesis.upnorte.edu.pe/bitstream/upnorte/193/1/Manuel%20Gonz%C3%A1lez%20y%20Luis%20V%C3%A1squez.pdf>, consultado el 21/04/2014.

Meneses, R. (2012). Disponible en <http://sistemas.uniandes.edu.co/~isis2603/dokuwiki/lib/exe/fetch.php?media=principal:isis2603-iso9000-3.pdf>, consultado el 21/04/2014.

NortiPC. (2013). Disponible en <http://norfipc.com/web/como-usar-html5-codigo-paginas-web-ventajas.html>, consultado el 24/04/2014.

Pérez, J. M. (2009). Universidad de Murcia. Disponible en <http://ocw.um.es/ingenierias/fundamentos-de-ingenieria-del-software/material-de-clase-1/capitulo06.pdf>, consultado el 28/04/2014.

Pública, M. d. (2014). Disponible en http://administracionelectronica.gob.es/pae_Home/pae_Documentacion/pae_Metodolog/pae_Metrica_v3.html#.U2F-JPldVg0, consultado el 30/04/2014.

Sánchez, D. L. (2013). Disponible en <http://prezi.com/qckqmue4wnxa/estandares-para-desarrollo-de-software/>, consultado el 03/05/2014.

Santamarina, M. S. (2011). Hypatia. Obtenido de revista de divulgación científico-tecnológica del gobierno del estado de Morelos. Disponible en <http://hypatia.morelos>

.gob.mx/index.php?option=com_content&task=view&id=711&Itemid=655, consultado el 03/05/2014.

Shneiderman, B. (2006.). Diseño de interfaces de usuario: estrategias para una interacción persona-computadora efectiva. Pearson/Addison Wesley.

Talens-Oliag, S. (2013). Instituto Tecnológico de Informática ITI. Disponible en <http://www.iti.es/media/about/docs/tic/11/articulo1.pdf>, consultado el 08/05/2014.

Untis Express. (2013). Disponible en <http://www.programahorario.com/index.php?lang=SP&gclid=Cl6xPTSorsCFdJcfgodhm4AbA>, consultado el 09/05/2014.

W3C. (2013). Disponible en <http://www.w3c.es/Divulgacion/GuiasBreves/HojasEstilo>, consultado el 12/05/2014.

World Wide Web consorcio. (2014). Disponible en <http://www.w3c.es/Divulgacion/GuiasBreves/Estandares>, consultado el 12/05/2014.

Anexos

En el CD donde se encuentra este trabajo se anexan lo documentos que se generaron durante el desarrollo de esta tesis.

