

UNIVERSIDAD POLITÉCNICA DE PUEBLA
Ingeniería en Informática



Proyecto de Estadía Profesional

“Sistema de Administración y Seguimiento de Proyectos”

Área temática del CONACYT: VII
Ingenierías y tecnologías

Presenta:

Josué Ammón Ortiz Gallegos

Asesor técnico

Adrián A. Cienfuegos Pérez

Asesor académico

Rebeca Rodríguez Huesca

Juan C. Bonilla, Puebla, México.

18 de diciembre de 2020

Resumen

La situación global actual y la creciente demanda de las Tecnologías de la Información es importante y relevante para los procesos productivos el poder hacer uso de herramientas informáticas que permitan realizar sus tareas de una forma más cómoda, ágil y accesible.

En el presente documento se presenta una de estas herramientas informáticas, un sistema de gestión de proyectos que permitirá a la empresa REDCOMM del Sureste S.A. de C.V. continuar realizando sus procesos de una forma más controlada y sin la necesidad de encontrarse todos en la misma oficina.

Se expone todo el proceso metodológico realizado paso por paso para llegar al fin de manera satisfactoria haciendo uso de la metodología de programación extrema que es ideal para proyecto con poca duración.

El sistema realizado es un sistema web haciendo uso del lenguaje php para la programación backend y usando lenguaje de etiquetas y estilos HTML y CSS dado que permiten tener el mismo sistema en distintos dispositivos con acceso a internet sin necesidad de realizar instalaciones adicionales en el dispositivo que se utilizará.

Índice

1. Introducción	6
1.1. Descripción del problema o necesidad	6
1.2 Justificación	7
1.3 Objetivo General y Específicos	7
2. Metodología y herramientas	9
2.1 Metodología Programación Extrema	9
2.2 Herramientas	15
3. Resultados	22
3.1 Planificación	22
3.2 Diseño	26
3.3 Codificación	30
3.4 Pruebas	37
3.5 Lanzamiento	39
4. Conclusiones y recomendaciones	40
5. Anexos	41
Apéndice A	41
Apéndice B	41
6. Referencias bibliográficas	43

Índice de figuras

Figura 1. Fases de la Metodología Programación Extrema	10
Figura 2. Prototipo de Inicio de sesión	28
Figura 3. Prototipo de pantalla principal	29
Figura 4. Prototipo de pantalla Información de cliente	29
Figura 5. Prototipo de pantalla usuarios (sólo administrador)	30
Figura 6. Pantalla de Inicio de sesión	32
Figura 7. Pantalla de usuarios (administrador).....	32
Figura 8. Pantalla agregar usuario	33
Figura 9. Pantalla modificar / eliminar usuario	33
Figura 10. Lista de clientes y proyectos en la pantalla principal	34
Figura 11. Frammento del formulario para añadir cliente	34
Figura 12. Pantalla de información de proyecto e historial de actividades	35
Figura 13. Formulario para agregar actividad	36
Figura 14. Formulario para generar informe.....	36

Índice de Tablas

Tabla 1. Historia de Usuario 1	22
Tabla 2. Historia de Usuario 2	23
Tabla 3. Historia de Usuario 3	23
Tabla 4. Historia de Usuario 4	23
Tabla 5. Historia de Usuario 5	24
Tabla 6. Historia de Usuario 6	24
Tabla 7. Historia de Usuario 7	24
Tabla 8. Historia de Usuario 8	25
Tabla 9. Historia de Usuario 9	25
Tabla 10. Planeación de iteraciones	26
Tabla 11. Tarjeta CRC Pantalla de Inicio de Sesión	27
Tabla 12. Tarjeta CRC Administrador de usuarios	27
Tabla 13. Tarjeta CRC Pantalla principal	27
Tabla 14. Tarjeta CRC Agregar Cliente	27
Tabla 15. Tarjeta CRC Agregar proyecto	27
Tabla 16. Tarjeta CRC Editar cliente	27
Tabla 17. Tarjeta CRC Información de cliente	27
Tabla 18. Tarjeta CRC Añadir evento	28
Tabla 19. Tarjeta CRC Informes	28
Tabla 20. Prueba de aceptación 01	37
Tabla 21. Prueba de aceptación 02	37
Tabla 22. Prueba de aceptación 03	38
Tabla 23. Prueba de aceptación 04	38
Tabla 24. Prueba de aceptación 05	39
Tabla 25. Prueba de aceptación 06	39

1. Introducción

En el siguiente capítulo se describe la problemática a abordar de manera que se pueda comprender correctamente la situación actual de la empresa con quien se desarrollará el proyecto, así como la solución propuesta.

1.1. Descripción del problema o necesidad

REDCOMM del Sureste S.A. de C.V. es una empresa de consultoría para optimización y ahorro de energía eléctrica, siendo fabricantes de sus propios equipos. Para generar una solución a las necesidades de los clientes, se genera un diagnóstico a través de la información que les proporciona. Esta información se consigue a través del área Comercial y del área de Operaciones y la integra el área de Proyectos. Con esta información se realiza el pre-proyecto que se destina a otra área para el desarrollo de la propuesta final, y en ocasiones el área comercial la presenta directamente al cliente, presentándose aquí un sesgo de comunicación con otras áreas.

Derivado de lo anterior se creó el área de Logística, que, en colaboración con el área de Proyectos, serán responsables del Procedimiento de Gestión de Proyectos, cuyo objetivo es resguardar, planificar, coordinar, controlar y dar seguimiento a las actividades y recursos necesarios para la gestión de los mismos.

Para el desarrollo de este procedimiento es indispensable contar con un sistema para resguardar, integrar y dar seguimiento a todas las etapas de esta gestión y que cada área involucrada tenga acceso a la información que le corresponda, identificando en todo momento en qué etapa se encuentra cada proyecto. Existe un plazo máximo para el cumplimiento de cada una de las etapas, para asegurar que se cumplan es necesario que se entrelacen las actividades y se generen “alertas” que permitan saber cuándo un proyecto termina una etapa y debe iniciar de manera inmediata la siguiente. Con esto se pretende eliminar esos sesgos de tiempo y comunicación.

El procedimiento delimita claramente las funciones y el alcance de cada área, en función de ello se establecerán los privilegios de los usuarios en el sistema para promover que se responsabilicen de las etapas que les corresponden de forma obligatoria y no exista ambigüedad respecto a sus actividades.

Por lo anterior se detectaron varias áreas de oportunidad en las que puede ayudar el sistema:

1. El resguardo integral de la información en un sistema.
2. Asegurar la secuencia del desarrollo del procedimiento.
3. Obligar a que cada área asuma sus funciones y alcances.

4. Mejorar la comunicación entre las áreas.
5. Conocer los tiempos de respuesta para mejorar la eficacia con el cliente.
6. Generar datos que permitan generar información estadística.
7. Acceder a la información en cualquier dispositivo y cualquier lugar con acceso a internet.
8. Al ser más eficientes en el tiempo de respuesta, las posibilidades de la venta aumentan.

De no atacarse estas áreas de oportunidad se podrían generar retrasos que se traducen en pérdidas económicas.

1.2 Justificación

Resulta de sumo valor el atacar la problemática encontrada dentro de la empresa, por lo que darle solución se traducirá en reducir la pérdida de ingresos ocasionada por proyectos que no tienen pronta respuesta.

Se desarrollará un sistema que permita:

- Agregar, modificar y consultar clientes
- Agregar, modificar y consultar proyectos
- Controlar y dar seguimiento a los proyectos según la información pertinente para cada área.
- Ingresar información según las responsabilidades de cada área
- Conocer en qué etapa se encuentra cada proyecto
- Establecer fechas límite para cada etapa
- Recibir “alertas” cuando una etapa termina y en caso de retrasos

Esto facilitará la obtención de datos que serán de utilidad para los encargados para generar indicadores claves y estadísticas que ayudarán a la empresa a identificar debilidades a atacar para generar estrategias que permitan reducir los tiempos de respuesta en cada etapa que se traducirán en más oportunidades de ventas al dar solución pronta a las necesidades de los clientes.

1.3 Objetivo General y Específicos

Objetivo general

Desarrollar un sistema que integre toda la información que generan las diferentes áreas involucradas en el procedimiento de gestión de proyectos para la empresa REDCOMM del Sureste S.A. de C.V.

Objetivos Específicos

- Analizar los procedimientos y las áreas relacionadas con la gestión de proyectos con el fin de concretar los pasos, roles e información que

manejará el sistema web para la correcta estructuración del sistema web.

- Diseñar la base de datos e interfaces del sistema web con el uso de diagramas entidad/relación y diagramas de caso de uso con el fin de concretar correctamente las interacciones y tareas que se realizarán en el sistema y la información que almacenará.
- Codificar el sistema web en base al diseño previamente realizado.
- Realizar pruebas de funcionalidad con ayuda de los usuarios finales que permitirán conocer si el sistema cumple con los requerimientos y expectativas o si es necesario realizar modificaciones previas a la entrega final.

2. Metodología y herramientas

En este capítulo se da una amplia explicación de las herramientas utilizadas, así como de sus ventajas y desventajas, también se expondrán los pasos de la metodología y lo que conlleva cada uno de ellos buscando dar explicación detallada de cómo nos beneficia esta para el desarrollo del proyecto.

2.1 Metodología Programación Extrema

XP o Programación Extrema es una metodología de desarrollo que pertenece a las conocidas como metodologías ágiles, cuyo objetivo es el desarrollo y gestión de proyectos con eficacia, flexibilidad y control.

Es un conjunto de técnicas que dan agilidad y flexibilidad en la gestión de proyectos. Se centra en crear un producto según los requisitos exactos del cliente. De ahí, que le involucre al máximo durante el método de gestión del desarrollo del producto. [1]

2.1.1 Características de la metodología:

Programación extrema es una de las metodologías más utilizadas en el desarrollo de software gracias a que posee características tales como:

- Comunicación constante entre el cliente y el equipo de desarrollo.
- Respuesta rápida a los cambios constantes.
- La planificación es abierta con un cronograma de actividades flexible.
- El software que funciona está por encima de cualquier otra documentación.
- Los requisitos del cliente y el trabajo del equipo del proyecto son los principales factores de éxito del mismo.

2.1.2 Fases:

Programación extrema pone el énfasis en la retroalimentación continua entre el cliente y el equipo de desarrollo y es idónea para proyectos con requisitos imprecisos y muy cambiantes.

La metodología de programación extrema se compone de cuatro fases previo al lanzamiento. Las cuales podemos ver representadas en la figura 1.



Figura 1. Fases de la Metodología Programación Extrema

2.1.2.1 Planificación

Durante la etapa de planificación se ven distintas actividades y factores que son de vital importancia para el desarrollo del sistema, las cuales son:

- **Historias de usuario.** El primer paso de cualquier proyecto que siga la metodología de Programación Extrema es definir las historias de usuario con el cliente. Las historias de usuario tienen la misma finalidad que los casos de uso, pero con lagunas diferencias: Constan de 3 o 4 líneas escritas por el cliente en un lenguaje no técnico sin hacer mucho hincapié en los detalles; no se debe hablar ni de posibles algoritmos para su implementación ni de diseños de base de datos adecuados, etc. Son usadas para estimar tiempos de desarrollo de la parte de la aplicación que describen. También se utilizan en la fase de pruebas, para verificar si el programa cumple con lo que especifica la historia de usuario. Cuando llega la hora de implementar una historia de usuario, el cliente y los desarrolladores se reúnen para concretar y detallar lo que tiene que hacer dicha historia. El tiempo de desarrollo ideal para una historia de usuario es entre 1 y 3 semanas.
- **Release planning.** Después de tener ya identificadas las historias de usuario es necesario crear un plan de publicaciones, en inglés “Release plan”, donde se indiquen las historias de usuario que se crearán para cada versión del programa y las fechas en las que se publicarán estas versiones. Es una planificación donde los desarrolladores y clientes establecen los tiempos de implementación ideales de las historias de usuario, la prioridad con la que serán implementadas y las historias que serán implementadas en cada versión del programa. Después de un “Release plan” tienen que estar claros estos cuatro factores: los objetivos que se deben cumplir (que son principalmente las historias que se deben desarrollar en cada versión), el tiempo que tardarán en desarrollarse y publicarse las versiones del programa, el número de personas que trabajarán en el desarrollo y cómo se evaluará la calidad del trabajo realizado.

- **Iteraciones.** Todo proyecto que siga la metodología de Programación Extrema se ha de dividir en iteraciones de aproximadamente 3 semanas de duración. Al comienzo de cada iteración los clientes deben seleccionar las historias de usuario definidas en el "Release planning" que serán implementadas. También se seleccionan las historias de usuario que no pasaron el test de aceptación que se realizó al terminar la iteración anterior. Estas historias de usuario son divididas en tareas de entre 1 y 3 días de duración que se asignarán a los programadores.
- **Velocidad de proyecto.** La velocidad del proyecto es una medida que representa la rapidez con la que se desarrolla el proyecto; estimarla es muy sencillo, basta con contar el número de las historias de usuario que se pueden implementar en una iteración; de esta forma, se sabrá el cupo de historias que se pueden desarrollar en las distintas iteraciones. Usando la velocidad del proyecto controlaremos que todas las tareas se puedan desarrollar en el tiempo del que dispone la iteración. Es conveniente reevaluar esta medida cada 3 o 4 iteraciones y si se aprecia que no es adecuada hay que negociar con el cliente un nuevo "Release Plan".
- **Programación en pareja.** La metodología aconseja la programación en parejas pues incrementa la productividad y la calidad del software desarrollado. El trabajo en pareja involucra a dos programadores trabajando en el mismo equipo; mientras uno codifica haciendo hincapié en la calidad de la función o método que está implementando, el otro analiza si ese método o función es adecuado y está bien diseñado. De esta forma se consigue un código y diseño con gran calidad.
- **Reuniones diarias.** Es necesario que los desarrolladores se reúnan diariamente y expongan sus problemas, soluciones e ideas de forma conjunta. Las reuniones tienen que ser fluidas y todo el mundo tiene que tener voz y voto.

2.1.2.2 Diseño

En este paso se intentará trabajar con un código sencillo, haciendo lo mínimo imprescindible para que funcione. Algunas de las actividades que se pueden realizar durante esta fase son:

- **Diseños Simples.** La metodología sugiere que hay que conseguir diseños simples y sencillos. Hay que procurar hacerlo todo lo menos complicado posible para conseguir un diseño fácilmente entendible e implementable que a la larga costará menos tiempo y esfuerzo desarrollar.
- **Glosarios de términos.** Usar glosarios de términos y una correcta especificación de los nombres de métodos y clases ayudará a comprender el diseño y facilitará sus posteriores ampliaciones y la reutilización del código.

- **Riesgos.** Si surgen problemas potenciales durante el diseño, la metodología sugiere utilizar una pareja de desarrolladores para que investiguen y reduzcan al máximo el riesgo que supone ese problema.
- **Funcionalidad extra.** Nunca se debe añadir funcionalidad extra al programa, aunque se piense que en un futuro será utilizada. Sólo el 10% de la misma es utilizada, lo que implica que el desarrollo de funcionalidad extra es un desperdicio de tiempo y recursos.
- **Refactorizar.** Refactorizar es mejorar y modificar la estructura y codificación de códigos ya creados sin alterar su funcionalidad. Refactorizar supone revisar de nuevo estos códigos para procurar optimizar su funcionamiento. Es muy común rehusar códigos ya creados que contienen funcionalidades que no serán usadas y diseños obsoletos. Esto es un error porque puede generar código completamente inestable y muy mal diseñado; por este motivo, es necesario refactorizar cuando se va a utilizar código ya creado.
- **Tarjetas CRC.** El uso de las tarjetas C.R.C. permiten al programador centrarse y apreciar el desarrollo orientado a objetos olvidándose de los malos hábitos de la programación procedural clásica. Las tarjetas C.R.C. representan objetos; la clase a la que pertenece el objeto se puede escribir en la parte de arriba de la tarjeta, en una columna a la izquierda se pueden escribir las responsabilidades u objetivos que debe cumplir el objeto y a la derecha, las clases que colaboran con cada responsabilidad.

2.1.2.3 Codificación

El cliente es una parte más del equipo de desarrollo; su presencia es indispensable en las distintas fases de Programación Extrema. A la hora de codificar una historia de usuario su presencia es aún más necesaria. No olvidemos que los clientes son los que crean las historias de usuario y negocian los tiempos en los que serán implementadas. Antes del desarrollo de cada historia de usuario el cliente debe especificar detalladamente lo que ésta hará y también tendrá que estar presente cuando se realicen los test que verifiquen que la historia implementada cumple la funcionalidad especificada.

La codificación debe hacerse atendiendo a estándares de codificación ya creados. Programar bajo estándares mantiene el código consistente y facilita su comprensión y escalabilidad.

Crear test que prueben el funcionamiento de los distintos códigos implementados nos ayudará a desarrollar dicho código. Crear estos test antes nos ayuda a saber qué es exactamente lo que tiene que hacer el código a implementar y sabremos que una vez implementado pasará dichos test sin problemas ya que dicho código ha sido diseñado para ese fin. Se puede dividir

la funcionalidad que debe cumplir una tarea a programar en pequeñas unidades, de esta forma se crearán primero los test para cada unidad y a continuación se desarrollará dicha unidad, así poco a poco conseguiremos un desarrollo que cumpla todos los requisitos especificados.

Como ya se comentó anteriormente, la metodología de programación extrema opta por la programación en pareja ya que permite un código más eficiente y con una gran calidad.

La metodología sugiere un modelo de trabajo usando repositorios de código dónde las parejas de programadores publican cada pocas horas sus códigos implementados y corregidos junto a los test que deben pasar. De esta forma el resto de programadores que necesiten códigos ajenos trabajarán siempre con las últimas versiones. Para mantener un código consistente, publicar un código en un repositorio es una acción exclusiva para cada pareja de programadores.

También propone un modelo de desarrollo colectivo en el que todos los programadores están implicados en todas las tareas; cualquiera puede modificar o ampliar una clase o método de otro programador si es necesario y subirla al repositorio de código. El permitir al resto de los programadores modificar códigos que no son suyos no supone ningún riesgo ya que para que un código pueda ser publicado en el repositorio tiene que pasar los test de funcionamiento definidos para el mismo.

La optimización del código siempre se debe dejar para el final. Hay que hacer que funcione y que sea correcto, más tarde se puede optimizar.

Programación Extrema afirma que la mayoría de los proyectos que necesiten más tiempo extra que el planificado para ser finalizados no podrán ser terminados a tiempo se haga lo que se haga, aunque se añadan más desarrolladores y se incrementen los recursos. La solución que plantea Programación Extrema es realizar un nuevo “Release plan” para concretar los nuevos tiempos de publicación y de velocidad del proyecto.

A la hora de codificar no seguimos la regla de programación extrema que aconseja crear test de funcionamiento con entornos de desarrollo antes de programar. Nuestros test los obtendremos de la especificación de requisitos ya que en ella se especifican las pruebas que deben pasar las distintas funcionalidades del programa, procurando codificar pensando en las pruebas que debe pasar cada funcionalidad.

2.1.2.4 Pruebas

Uno de los pilares de la metodología de Programación Extrema es el uso de test para comprobar el funcionamiento de los códigos que vayamos implementando.

Se debe crear las aplicaciones que realizarán los test con un entorno de desarrollo específico para test.

Hay que someter a tests las distintas clases del sistema omitiendo los métodos más triviales.

Se deben crear los test que pasarán los códigos antes de implementarlos; en el apartado anterior se explicó la importancia de crear antes los test que el código.

Un punto importante es crear test que no tengan ninguna dependencia del código que en un futuro evaluará. Hay que crear los test abstrayéndose del futuro código, de esta forma aseguraremos la independencia del test respecto al código que evalúa.

Como se comentó anteriormente los distintos test se deben subir al repositorio de código acompañados del código que verifican. Ningún código puede ser publicado en el repositorio sin que haya pasado su test de funcionamiento, de esta forma, aseguramos el uso colectivo del código (explicado en el apartado anterior).

El uso de los test es adecuado para observar la refactorización. Los test permiten verificar que un cambio en la estructura de un código no tiene por qué cambiar su funcionamiento.

Test de aceptación. Los test mencionados anteriormente sirven para evaluar las distintas tareas en las que ha sido dividida una historia de usuario. Para asegurar el funcionamiento final de una determinada historia de usuario se deben crear "Test de aceptación"; estos test son creados y usados por los clientes para comprobar que las distintas historias de usuario cumplen su cometido.

Al ser las distintas funcionalidades de nuestra aplicación no demasiado extensas, no se harán test que analicen partes de las mismas, sino que las pruebas se realizarán para las funcionalidades generales que debe cumplir el programa especificado en la descripción de requisitos.

2.1.2.5 Lanzamiento

Llegar a este punto significa que se han probado todas las historias de usuario o mini-versiones con éxito, ajustándose a los requerimientos del cliente. Tenemos un software útil y podemos incorporarlo en el producto.[1]

2.1.3 Ventajas:

- Programación organizada.
- Menor tasa de errores.
- Satisfacción del programador.

2.1.4 Desventajas:

- Es recomendable emplearlo sólo en proyectos a corto plazo.
- Altas comisiones en caso de fallar.

2.2 Herramientas

En esta sección se presentan las herramientas seleccionadas para el desarrollo web del proyecto, también se dirán las ventajas y desventajas de las mismas.

2.2.1 HTML5

Por sus siglas en inglés *HyperText Markup Language*, HTML no es un lenguaje de programación sino un lenguaje de marcado de hipertexto, se escribe en su totalidad con elementos, estos elementos están constituidos por etiquetas, contenido y atributos.

Es un lenguaje que interpreta el navegador web para mostrar los sitios o aplicaciones web tal y como estamos acostumbrados.[2]

2.2.1.1 Ventajas

- Texto presentando de forma estructurada y agradable.
- Archivos pequeños.
- Despliegue rápido.
- Compatibilidad con todos los navegadores web.

2.2.1.2 Desventajas

- Lenguaje estático.
- La interpretación de cada navegador puede ser diferente.
- Guarda muchas etiquetas que pueden convertirse en “basura” y dificultan la corrección.
- El diseño es más lento.

- Las etiquetas son muy limitadas.

2.2.2 PHP

Hypertext Preprocessor por su acrónimo en inglés, es un lenguaje de código abierto muy popular especialmente adecuado para el desarrollo web y que puede ser incrustado en HTML.

En lugar de usar muchos comandos para mostrar HTML, las páginas de PHP contienen HTML con código incrustado. El código PHP está encerrado entre las etiquetas especiales de comienzo y final que permiten entrar y salir del “modo PHP”.

Este código se distingue de otros como JavaScript ya que es ejecutado en el servidor, generando HTML y enviándolo al cliente. El cliente recibirá el resultado de ejecutar el script, aunque no se sabrá el código subyacente que era. El servidor web puede ser configurado incluso para que procese todos los ficheros HTML con PHP, por lo que no hay manera de que los usuarios puedan saber qué se tiene debajo de la manga.[3]

2.2.2.1 Ventajas

- Lenguaje totalmente libre y abierto.
- Posee una curva de aprendizaje muy baja.
- Los entornos de desarrollo son de rápida y fácil configuración.
- Fácil de instalar.
- Fácil acceso e integración con las bases de datos.
- Posee una comunidad muy grande.
- El lenguaje con mayor usabilidad en el mundo.
- Es un lenguaje multiplataforma.
- Completamente orientado al desarrollo de aplicaciones web dinámicas y/o páginas web con acceso a una Base de Datos.
- El código escrito en PHP es invisible al navegador ya que se ejecuta al lado del servidor y los resultados en el navegador es HTML.
- Posee una versatilidad para la conexión con la mayoría de bases de datos que existen en la actualidad.

2.2.2.2 Desventajas

- El inconveniente es que el código fuente no pueda ser ocultado de una manera eficiente. La ofuscación es una técnica que puede dificultar la lectura del código y, en ciertos aspectos representa tiempos de ejecución.

- Nuestro código estará seguro para ejecutar si es nuestro propio servidor. Por lo tanto, si un cliente requiere su código en su pc, tendríamos que dejar el código fuente, sin manera de ocultarlo, aunque hay muchas aplicaciones para PHP que nos ayuda a encriptar el código fuente.
- Si no lo configuras y/o proteges correctamente dejas abiertas muchas brechas de seguridad que a la larga tendremos problemas.
- Sólo se ejecuta en un servidor y se necesita un servidor web para que funcione.

2.2.3 JavaScript

A diferencia de PHP, JavaScript funciona del lado del cliente, los navegadores son los encargados de interpretar estos códigos.

Tiene la ventaja de ser incorporado en cualquier página web, puede ser ejecutado sin la necesidad de instalar otro programa para ser visualizado.

Este lenguaje está basado en acciones que posee menos restricciones, gran parte de la programación de este lenguaje está centrada en describir objetos, escribir funciones que respondan a movimientos del mouse, aperturas, utilización de teclas, cargas de páginas entre otros.

Existen dos tipos de JavaScript:

1. Navigator JavaScript (del lado del cliente)
2. LiveWire JavaScript (del lado del servidor) [4]

2.2.3.1 Ventajas

- Es un lenguaje muy sencillo.
- Es rápido, por lo tanto, tiende a ejecutar las funciones inmediatamente.
- Cuenta con múltiples opciones de efectos visuales.
- Es soportado por los navegadores más populares y es compatible con los dispositivos más modernos.
- Es muy versátil, puesto que es muy útil para desarrollar páginas dinámicas y aplicaciones web,
- Es una buena solución para poner en práctica la validación de datos en un formulario.
- Es una multiplataforma, puede ser ejecutado de manera híbrida en cualquier sistema operativo móvil.
- Es el único lenguaje que permite trabajar modo “*fullstack*” en cualquier tipo de desarrollo de programación.

2.2.3.2 Desventajas

- En el “*frontend*” sus códigos son visibles, por lo tanto, pueden ser leídos por cualquier usuario.
- Tiende a introducir gran cantidad de fragmentos de código en los sitios web.
- Sus opciones 3D son limitadas, si se quiere utilizar este lenguaje de programación para crear un juego, debe emplearse otras herramientas.
- No es compatible en todos los navegadores de manera uniforme.
- Los usuarios tienen la opción de desactivar JavaScript desde su navegador.
- Sus scripts son limitados por razones de seguridad y no es posible realizar todo con JavaScript, por lo tanto, es necesario complementarlo con otros lenguajes evolucionados y más seguros.

2.2.4 CSS

CSS es un lenguaje de hojas de estilos creado para controlar el aspecto o presentación de los documentos electrónicos definidos con HTML y XHTML. CSS es la mejor forma de separar los contenidos y su presentación y es imprescindible para crear páginas web complejas.

Separar la definición de los contenidos y la definición de su aspecto presenta numerosas ventajas, ya que obliga a crear documentos HTML/XHTML bien definidos y con significado completo (también llamados "documentos semánticos"). Además, mejora la accesibilidad del documento, reduce la complejidad de su mantenimiento y permite visualizar el mismo documento en infinidad de dispositivos diferentes.

Al crear una página web, se utiliza en primer lugar el lenguaje HTML/XHTML para marcar los contenidos, es decir, para designar la función de cada elemento dentro de la página: párrafo, titular, texto destacado, tabla, lista de elementos, etc.

Una vez creados los contenidos, se utiliza el lenguaje CSS para definir el aspecto de cada elemento: color, tamaño y tipo de letra del texto, separación horizontal y vertical entre elementos, posición de cada elemento dentro de la página, etc. [5]

2.2.4.1 Ventajas

- Con una Hoja de Estilo podemos alterar la presentación de cada elemento el código HTML, ahorrando esfuerzo y tiempo de edición. De

este modo no sólo simplificamos el mantenimiento del sitio, sino que además reducimos las posibilidades de cometer errores.

- El lenguaje de las CSS ofrece herramientas de composición más potentes que HTML. Más aún, las CSS permiten aplicar prácticamente todas las propiedades a cualquier elemento de la página, mientras que HTML sólo permite un número limitado de propiedades para cada elemento.
- Se evita tener que recurrir a trucos para conseguir algunos efectos. Con CSS no es necesario usar imágenes invisibles para hacer una sangría o usar una tabla para ubicar un elemento en determinado lugar de la pantalla (las CSS permiten posicionar con precisión cualquier elemento).
- El lenguaje de las Hojas de Estilo, aunque muy potente, es relativamente sencillo y fácil de aprender.
- Los documentos que usan CSS generalmente resultan más compactos.
- Las Hojas de Estilo pueden aplicarse de varias maneras y combinarse formando una cascada de estilos con la información de cada una.
- Pueden usarse con otros lenguajes de programación (como JavaScript) para conseguir efectos dinámicos en las páginas.
- Se pueden especificar Hojas de Estilo para diferentes navegadores y tipos de medios (impresos, braille, auditivos, etc.).

2.2.4.2 Desventajas

- Soporte irregular entre los distintos navegadores.
- Existen diferencias entre cada navegador e inclusive entre distintas versiones.
- Contenido inaccesible por incompatibilidad.

2.2.5 Bootstrap

Bootstrap es un framework CSS desarrollado por Twitter en 2010, para estandarizar las herramientas de la compañía.

Inicialmente, se llamó Twitter Blueprint y, un poco más tarde, en 2011, se transformó en código abierto y su nombre cambió para Bootstrap. Desde entonces fue actualizado varias veces y ya se encuentra en la versión 4.4.

El framework combina CSS y JavaScript para estilizar los elementos de una página HTML. Permite mucho más que, simplemente, cambiar el color de los botones y los enlaces.

Esta es una herramienta que proporciona interactividad en la página, por lo que ofrece una serie de componentes que facilitan la comunicación con el usuario, como menús de navegación, controles de página, barras de progreso y más.

Además de todas las características que ofrece el framework, su principal objetivo es permitir la construcción de sitios web responsive para dispositivos móviles.

Esto significa que las páginas están diseñadas para funcionar en desktop, tablets y smartphones, de una manera muy simple y organizada. [6]

2.2.5.1 Ventajas

- Utiliza componentes y servicios creados por la comunidad web, tales como: HTML5 shim, Normalize.css, OOCSS (CSS orientado a objetos), jQuery UI, LESS y GitHub.
- Es un conjunto de buenas prácticas
- El famoso Grid system, que por defecto incluye 12 columnas fijas o fluidas, dependiendo de si tu diseño será Responsive o no. Está orientado siguiente el concepto Mobile First.
- Soporte: Hay una enorme comunidad que soporta este desarrollo y cuenta con implementaciones externas como WordPress, Drupal, SASS o jQuery UI.
- Comodidad y rapidez: Herramienta sencilla y ágil para construir sitios web e interfaces.
- Componentes: Mucha variedad de plantillas y temas.

2.2.5.2 Desventajas

- Aprendizaje: Es necesario adaptarse a su forma de trabajo, si bien su curva de aprendizaje es liviana, deberás comprender y familiarizarte con su estructura y nomenclatura.
- Adaptación: Debes adaptar tu diseño a un grid de 12 columnas, que se modifican según el dispositivo. Aquí empiezan los problemas, Bootstrap por defecto te trae anchos, márgenes y altos de línea, y realizar cambios específicos es por decir, un poco tedioso.
- Mantenimiento: Es complicado, cambiar de versión si has realizado modificaciones profundas sobre el núcleo.
- Ampliar componentes: Si necesitas añadir componentes que no existen, debes hacerlos tú mismo en CSS y cuidar de que mantenga coherencia con tu diseño y cuidando el responsive.
- Pesado: No es ligero, y además, para algunas funcionalidades, será necesario tener que usar JavaScript y jQuery.

2.2.6 MySQL

MySQL es un sistema de gestión de bases de datos (SGBD) SQL que en algunos aspectos es aproximadamente tan potente como Oracle. Cabe mencionar que, a mediados del año 2009, Oracle, ha adquirido MySQL.

Sus principales objetivos han sido la velocidad y la robustez. Es un SGBD sencillo y rápido que se adapta perfectamente a entornos en los que el volumen de datos sea del orden de megabytes. En la versión 5 de MySQL. Ha incluido el control de transacciones, procedimientos almacenados y triggers, por lo que ha rellenado el gran hueco que lo diferenciaba de grandes SGBD como Oracle. Si bien existe la posibilidad de comprar su soporte, Oracle sigue teniendo más aceptación en el mundo empresarial. [7]

2.2.6.1 Ventajas

- MySQL es de código abierto
- Velocidad al realizar las operaciones, lo que le hace uno de los gestores con mejor rendimiento.
- Bajo costo en requerimientos para la elaboración de bases de datos, ya que debido a su bajo consumo puede ser ejecutado en una máquina con escasos recursos sin ningún problema.
- Facilidad de configuración e instalación.
- Soporta gran variedad de sistemas operativos.
- Baja probabilidad de corromper datos, incluso si los errores no se producen en el propio gestor, sino en el sistema en el que está.
- Su conectividad, velocidad y seguridad hacen de MySQL Server altamente apropiado para acceder a bases de datos en internet.

2.2.6.2 Desventajas

- Un gran porcentaje de las utilidades de MySQL no están documentadas.
- No es intuitivo, como otros programas.

3. Resultados

En este capítulo se explica y describe de forma detallada la forma en la que se ejecutó el proyecto en base a la metodología mencionada en el capítulo anterior.

3.1 Planificación

Como parte de la planificación se expone a continuación las historias de usuario obtenidas y el Release Planning del proyecto que serán la guía para todo el desarrollo.

3.1.1 Historias de usuario

De acuerdo a las reuniones realizadas y al análisis hecho dentro de la empresa se pudieron identificar las siguientes historias de usuario que a su vez servirán como guía para poder realizar el sistema. Las historias de usuario se encuentran expuestas en las tablas 1 a la 7.

Historias de Usuario	
Número: 1	Usuario: Todos los Usuarios
Nombre de Historia: Inicio de sesión	
Prioridad: Alta	Riesgo en desarrollo: Alto
Puntos estimados: 3	Iteración asignada: 1
Programador responsable: Josué Ammón Ortiz Gallegos	
Descripción: Como: Administrador, directivo, gestor de proyectos, logística, producción y vendedor Quiero: Ingresar correo electrónico y contraseña Para: Acceder al sistema	
Validación: Que el usuario pueda ingresar su correo electrónico y contraseña para obtener acceso a las funciones correspondientes y que no le permita ingresar en caso de ingresar datos erróneamente.	

Tabla 1. Historia de Usuario 1

Historias de Usuario	
Número: 2	Usuario: Administrador
Nombre de Historia: Administrar usuarios	
Prioridad: Media	Riesgo en desarrollo: Alto
Puntos estimados: 5	Iteración asignada: 1
Programador responsable: Josué Ammón Ortiz Gallegos	
Descripción: Como: Administrador Quiero: Tener acceso a una lista de los usuarios del sistema, poder agregar nuevos usuarios, así como modificar o eliminar a los ya existentes Para: Gestionar y controlar quiénes pueden acceder al sistema	
Validación: Sólo le aparece esta pantalla al administrador, se pueden agregar, modificar y	

eliminar usuarios correctamente, en caso de modificar datos se puede sólo modificar alguno o todos los datos.

Tabla 2. Historia de Usuario 2

Historias de Usuario	
Número: 3	Usuario: Todos los Usuarios
Nombre de Historia: Mostrar clientes y sus proyectos	
Prioridad: Alta	Riesgo en desarrollo: Alto
Puntos estimados: 5	Iteración asignada: 2
Programador responsable: Josué Ammón Ortiz Gallegos	
<p>Descripción:</p> <p>Como: Administrador, directivo, gestor de proyectos, logística, producción y vendedor</p> <p>Quiero: Observar los clientes a los que tengo acceso, así como sus respectivos proyectos o ver si no tienen proyectos</p> <p>Para: Conocer el estado del proyecto de cada cliente y acceder a información detallada desde la pantalla principal</p>	
<p>Validación:</p> <p>Se muestran los clientes y proyectos filtrados según el nivel de usuario que inició el sistema.</p>	

Tabla 3. Historia de Usuario 3

Historias de Usuario	
Número: 4	Usuario: Todos los Usuarios
Nombre de Historia: Agregar cliente	
Prioridad: Alta	Riesgo en desarrollo: Medio
Puntos estimados: 3	Iteración asignada: 3
Programador responsable: Josué Ammón Ortiz Gallegos	
<p>Descripción:</p> <p>Como: Administrador, directivo, gestor de proyectos, logística, producción y vendedor</p> <p>Quiero: Generar un formulario digital basado en el actual "Formato de Información Comercial"</p> <p>Para: Tener un registro digital de todos los clientes con los mismos datos que se utilizan actualmente</p>	
<p>Validación:</p> <p>Al llenar el formulario de agregar cliente este aparecerá reflejado en su lista de clientes sin ningún proyecto asignado en la pantalla principal</p>	

Tabla 4. Historia de Usuario 4

Historias de Usuario	
Número: 5	Usuario: Administrador y Gestor de Proyectos
Nombre de Historia: Agregar Proyecto	
Prioridad: Alta	Riesgo en desarrollo: Medio
Puntos estimados: 5	Iteración asignada: 3
Programador responsable: Josué Ammón Ortiz Gallegos	
<p>Descripción:</p> <p>Como: Administrador y Gestor de Proyectos</p> <p>Quiero: Agregar un proyecto a un cliente existente escogiendo el tipo de servicio y agregando una breve descripción</p>	

Para: Tener un registro de los proyectos existentes y poder realizarles seguimiento
Validación: Se puede agregar un proyecto exitosamente a un cliente haciendo uso de cualquiera de las dos maneras existentes y aparece el cliente con sus proyectos creados en la tabla de clientes mostrada en la pantalla principal

Tabla 5. Historia de Usuario 5

Historias de Usuario	
Número: 6	Usuario: Vendedor / Usuario ligado al cliente
Nombre de Historia: Editar Cliente	
Prioridad: Media	Riesgo en desarrollo: Medio
Puntos estimados: 3	Iteración asignada: 4
Programador responsable: Josué Ammón Ortiz Gallegos	
Descripción: Como: Administrador, directivo, gestor de proyectos, logística, producción y vendedor creador del cliente Quiero: Editar la información del cliente en caso de ser necesario Para: Mantener la información de mi cliente actualizada en el sistema	
Validación: La información es mostrada correctamente y al realizar cambios son actualizados satisfactoriamente	

Tabla 6. Historia de Usuario 6

Historias de Usuario	
Número: 7	Usuario: Todos los Usuarios
Nombre de Historia: Consultar Información de cliente	
Prioridad: Alta	Riesgo en desarrollo: Medio
Puntos estimados: 5	Iteración asignada: 4
Programador responsable: Josué Ammón Ortiz Gallegos	
Descripción: Como: Administrador, directivo, gestor de proyectos, logística, producción y vendedor Quiero: Acceder a la información del cliente que le corresponda a mi nivel de usuario, así como al historial de las actividades que se han realizado con respecto a su proyecto Para: Poder conocer el estado del proyecto y su progreso	
Validación: La pantalla muestra información filtrada según el nivel de usuario y el historial de eventos	

Tabla 7. Historia de Usuario 7

Historias de Usuario	
Número: 8	Usuario: Producción, Gestión de Proyectos, Directivos
Nombre de Historia: Añadir Evento	
Prioridad: Alta	Riesgo en desarrollo: Alta
Puntos estimados: 20	Iteración asignada: 5
Programador responsable: Josué Ammón Ortiz Gallegos	
Descripción: Como: Directivo, Gestor de proyectos y producción Quiero: Añadir eventos a un proyecto, eligiendo el tipo de actividad que se	

<p>realizó, permitiéndome agregar una breve descripción de lo que se realizó, adjuntar un archivo, modificar el estado del proyecto en caso de ser necesario y elegir al encargado de la siguiente actividad notificándole por correo electrónico</p> <p>Para: Agregar información al historial de actividades, mantener la información actualizada</p>
<p>Validación:</p> <p>Se agrega una actividad realizada, se guarda el archivo en el servidor, aparece la actividad en el historial de actividades del proyecto y el usuario que recibe para la siguiente actividad recibe una notificación vía email</p>

Tabla 8. Historia de Usuario 8

Historias de Usuario	
Número: 9	Usuario: Gestión de proyectos, Administrador, Directivos
Nombre de Historia: Realizar Informes	
Prioridad: Media	Riesgo en desarrollo: Bajo
Puntos estimados: 13	Iteración asignada: 6
Programador responsable: Josué Ammón Ortiz Gallegos	
<p>Descripción:</p> <p>Como: Administrador, Directivo y Gestor de proyectos</p> <p>Quiero: Realizar informes escogiendo el tipo de datos que quiero en él, el periodo de tiempo y el formato del archivo ya sea PDF o XSLM</p> <p>Para: Poder evaluar los procesos relacionados con la gestión de proyectos y usarlos como apoyo a la toma de decisiones dentro de la empresa</p>	
<p>Validación:</p> <p>Se puede escoger entre informes de clientes, generarlo y descargarlo</p>	

Tabla 9. Historia de Usuario 9

3.1.2 Release Planning

Una vez establecidas las historias de usuario a realizar, se generó una planeación de salida de versiones donde se irían incorporando cada una de las historias de usuario como lo podremos observar en la tabla 10.

Iteración	Historia de Usuario	Objetivo	Tiempo de desarrollo
1	Pantalla de Inicio de sesión	Desarrollar una interfaz funcional de inicio de sesión que solicite correo electrónico y contraseña	1 semana
1	Administrar Usuarios	Desarrollar interfaces funcionales que muestren los usuarios existentes, que permita modificar, añadir y eliminar usuarios	1 semanas
2	Pantalla principal	Desarrollar una interfaz funcional que muestre los clientes existentes y sus proyectos según el área al que pertenezca el usuario	2 semanas
3	Agregar Clientes	Desarrollar un formulario digital basándose en el actual documento "información comercial" donde se registran los datos necesarios para	1 semanas

		la empresa con respecto a un cliente	
3	Agregar Proyecto	Desarrollar un formulario donde se ingrese el tipo de proyecto a desarrollar y se ingrese información adicional relevante para el proyecto	1 semana
4	Editar Cliente	Crear una pantalla donde se podrá editar la información del cliente	1 semana
4	Información de Cliente	Mostrar información del cliente según el nivel de usuario, así como un historial de eventos	1 semanas
5	Añadir Evento	Crear una pantalla que permitirá registrar una actividad por tipo de actividad, añadir una descripción y subir archivos de ser necesario	2 semanas
6	Informes	Realizar una pantalla donde se podrá escoger el tipo de informe a crear, periodo de tiempo y formato	

Tabla 10. Planeación de iteraciones

3.2 Diseño

Como parte del diseño se hizo uso de tarjetas CRC para facilitar el modelado de las funciones que se llevarán a cabo en el sistema como se muestran a continuación. Además, se muestran los diseños simples que son prototipos de baja fidelidad que ayudan a agilizar la codificación de la interfaz.

3.2.1 Tarjetas CRC

En el caso de este sistema, las tarjetas CRC se utilizaron para poder definir roles del inicio de sesión y definir las actividades que podrán realizar cada tipo de usuario en el sistema dado que cada uno tiene distintas funciones y un máximo de información que puede observar o modificar. En las tablas de la 11 a la 19 podremos observar las tarjetas CRC realizadas en base a las Historias de Usuario.

Pantalla de Inicio de Sesión	
<ul style="list-style-type: none"> • Recibir credenciales de inicio de sesión • Validar que los datos sean del formato correcto • Buscar credenciales en base de datos • Redirigir a pantalla principal y abrir sesión en caso de recibir credenciales encontradas en la base de datos • Redirigir al inicio de sesión en caso de recibir credenciales incorrectas 	<ul style="list-style-type: none"> • Administrador de Usuarios

Tabla 11. Tarjeta CRC Pantalla de Inicio de Sesión

Administrador de Usuarios	
<ul style="list-style-type: none"> • Añadir Usuarios • Eliminar Usuarios • Modificar Usuarios 	

Tabla 12. Tarjeta CRC Administrador de usuarios

Pantalla principal	
<ul style="list-style-type: none"> • Agregar clientes • Visualizar lista de clientes • Visualizar proyectos de sus clientes 	<ul style="list-style-type: none"> • Administrador de usuarios • Pantalla inicio de sesión

Tabla 13. Tarjeta CRC Pantalla principal

Agregar cliente	
<ul style="list-style-type: none"> • Recibir los datos generales del cliente a ingresar • Validar consistencia de datos ingresados 	<ul style="list-style-type: none"> • Pantalla inicio de sesión • Pantalla principal

Tabla 14. Tarjeta CRC Agregar Cliente

Agregar proyecto	
<ul style="list-style-type: none"> • Elegir tipo de servicio al que pertenecerá • Recibir una descripción breve al proyecto 	<ul style="list-style-type: none"> • Agregar cliente • Pantalla inicio de sesión • Administrador de Usuarios

Tabla 15. Tarjeta CRC Agregar proyecto

Editar cliente	
<ul style="list-style-type: none"> • Recibir los datos actuales del cliente de la base de datos • Modificar datos de la base de datos de ser requerido 	<ul style="list-style-type: none"> • Agregar cliente • Pantalla inicio de sesión • Administrador de usuarios

Tabla 16. Tarjeta CRC Editar cliente

Información de cliente	
<ul style="list-style-type: none"> • Mostrar datos del cliente según su nivel de usuario • Mostrar historial de actividades del proyecto del usuario • Mostrar botón para agregar actividad según su nivel de usuario 	<ul style="list-style-type: none"> • Agregar cliente • Pantalla inicio de sesión • Administrador de usuarios • Agregar proyecto

Tabla 17. Tarjeta CRC Información de cliente

Añadir evento	
<ul style="list-style-type: none"> • Elegir tipo de evento • Añadir descripción del evento • Añadir archivo adjunto (opcional) 	<ul style="list-style-type: none"> • Pantalla inicio de sesión • Administrador de usuarios • Agregar cliente • Agregar proyecto

<ul style="list-style-type: none"> • Elegir usuario que recibirá para la actividad 	<ul style="list-style-type: none"> • Información de cliente
---	--

Tabla 18. Tarjeta CRC Añadir evento

Informes	
<ul style="list-style-type: none"> • Elegir tipo de informe • Elegir periodo de tiempo • Elegir formato de descarga • Generar informe especificado 	<ul style="list-style-type: none"> • Pantalla inicio de sesión • Administrador de usuarios

Tabla 19. Tarjeta CRC Informes

3.2.2 Diseños Simples

Los diseños simples permiten ampliar el panorama y pulir la idea al decidir cómo acomodar cada una de las funciones del sistema.

Los diseños simples de este proyecto fueron realizados utilizando la herramienta Balsamiq dado que permite crear un prototipo de baja fidelidad de manera simple y más cercana a la realidad que en conjunto con el desarrollo empleando Bootstrap agilizaron el proceso de diseño de las pantallas a utilizar. En las figuras 2 a la 5 podremos observar algunas de las pantallas diseñadas por medio de la herramienta ya mencionada.



Figura 2. Prototipo de Inicio de sesión

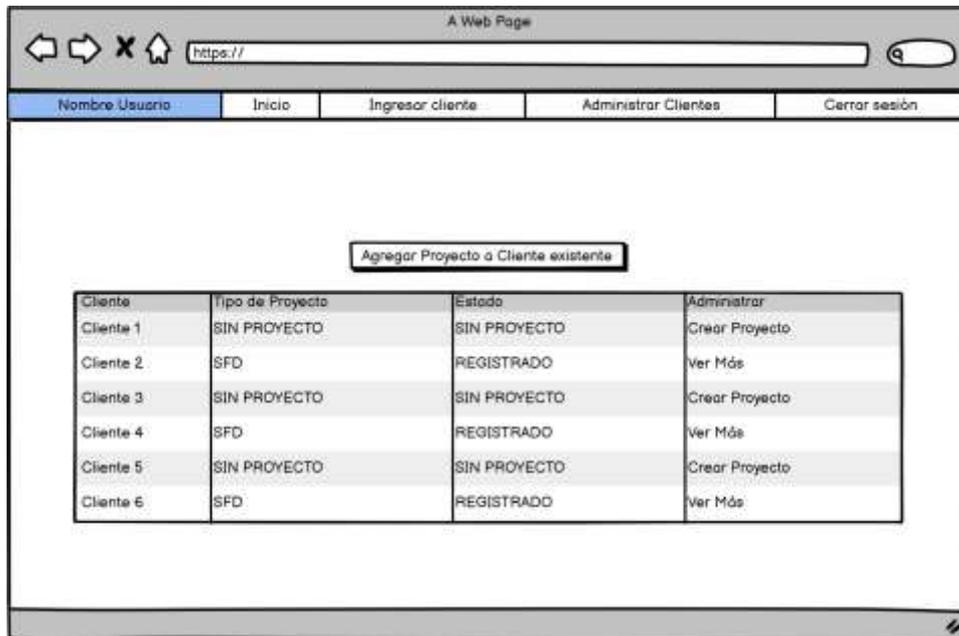


Figura 3. Prototipo de pantalla principal

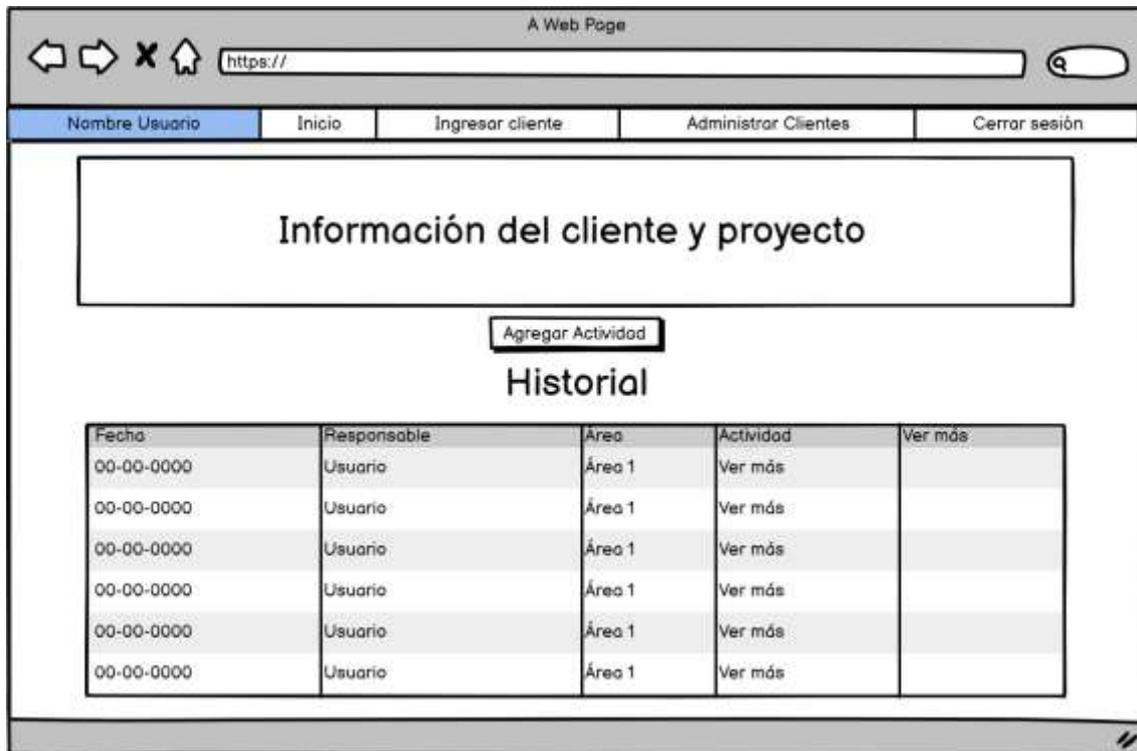


Figura 4. Prototipo de pantalla Información de cliente



Figura 5. Prototipo de pantalla usuarios (sólo administrador)

3.3 Codificación

A continuación, se describe la codificación realizada para el sistema dividido por iteraciones, se explica cada módulo desarrollado por iteración de manera breve.

3.3.1 Preparación de la plantilla web

Dado que todas las ventanas hacen uso de Bootstrap como Framework de CSS todas las ventanas deberán tener el mismo código inicial entonces se creó una plantilla que fue utilizada en todos los casos de pantallas que incluyen:

- Etiquetas link, script y style pertenecientes al Framework Bootstrap y al CSS del sistema
- Etiquetas pertenecientes a la barra de navegación que estará presente en todas las pantallas

Para poder ver parte del código plantilla véase en el capítulo 5. Anexos en el inciso A.

3.3.2 Creación de base de datos

La base de datos fue creada haciendo uso de phpMyAdmin que brinda una interfaz gráfica al gestor de bases de datos de MySQL lo cual simplifica el diseño y codificación de la misma.

La base de datos fue creciendo progresivamente según las necesidades de los módulos en el siguiente orden:

1. Usuarios
2. Estados (para facilitar la creación de códigos de proyecto)
3. Clientes
4. Servicio (para tener una lista de servicios)
5. Proyectos
6. T_Actividad (para estandarizar el tipo de actividades del historial)
7. Actividad

De esta forma se podían ir probando funciones mientras continuaba el desarrollo del sistema.

3.3.3 Código de acceso a la base de datos

Para facilitar de la misma forma el acceso a la base de datos sin necesidad de repetir la sentencia de conexión se creó un archivo denominado “conexión.php” el cual contiene las sentencias principales de conexión a la base de datos haciendo uso del Framework PDO que adicionalmente elimina las posibilidades de sufrir ataques de inyección SQL. Para ver el código de conexión véase en el capítulo 5. Anexos en el inciso B.

3.3.4 Desarrollo de la primera iteración

Lo primero que se realizó fue el desarrollo del código HTML con su respectivo CSS, una vez finalizada la interfaz se procedió a codificar las interacciones, se utilizó el código ya mencionado para la conexión con la base de datos y se hace uso del algoritmo md5 para encriptación de datos que viene integrado con php.

Una vez comprobado que el usuario es correcto se procede a tomar los datos necesarios de la base de datos tales como, nombre, id de usuario y nivel de usuario. Estos datos son llevados a través de las distintas pantallas del sistema por medio de la propiedad “session” que permitirá trasladar datos relevantes entre las distintas ventanas que la contengan, así como mantener una sesión iniciada aun cuando el navegador de internet sea cerrado. La pantalla de inicio de sesión se muestra en la figura 6.

Inicio de Sesión

Correo Electrónico

Contraseña

Iniciar Sesión

Figura 6. Pantalla de Inicio de sesión

Ya creado el inicio de sesión se procedió a realizar el módulo que permitirá al administrador agregar, modificar y eliminar usuarios existentes. Para ello se comprueba que el nivel de usuario sea el correcto, una vez que el nivel del usuario se comprobó muestra la lista de los usuarios existentes en una tabla llenada por medio de una secuencia SQL, así como un botón para modificar y un botón para añadir un nuevo usuario como se muestra en la figura 7.

Lista de Usuarios

Agregar

#	Nombre	Editar
1	Josué Ammón Ortiz Gallegos	Editar

Figura 7. Pantalla de usuarios (administrador)

En la pantalla de añadir usuario se piden todos los datos necesarios para el funcionamiento del resto del sistema y en la pantalla de modificar muestra todos los datos ya introducidos excepto la contraseña permitiendo que esta pueda ser modificada en caso de así desearlo como se muestra en las figuras 8 y 9.

Agregar Usuario

Nombre:

Correo Electrónico:

Nivel de Usuario:

Teléfono:

Contraseña:

CREAR

Figura 8. Pantalla agregar usuario

Editar Usuario

Nombre:

Correo Electrónico:

Nivel de Usuario:

Teléfono:

Contraseña:

ACTUALIZAR **ELIMINAR**

Figura 9. Pantalla modificar / eliminar usuario

3.3.5 Desarrollo de la segunda iteración

La segunda iteración se centró directamente en la pantalla principal que todos los usuarios verán, la información que se muestra en esta pantalla depende del nivel de usuario con el cual se accede. El usuario que verá reducida la información en esta pantalla es el vendedor, puesto que su principal función es añadir nuevos clientes al sistema, fuera de eso, el vendedor solamente necesita acceso a sus propios clientes para poder modificar datos relevantes como se observa en la figura 10.

Cliente	Tipo de Proyecto	Estado	Administrar
ejemplo 1	Sistemas Fotovoltaicos Domésticos	REGISTRADO	Ver Más
ejemplo 1	Sophia	REGISTRADO	Ver Más
Cliente del vendedor 1	Generación Distribuida Industria	REGISTRADO	Ver Más

Figura 10. Lista de clientes y proyectos en la pantalla principal

Por otro lado, los demás tipos de usuarios necesitarán acceso a todos los clientes creados en el sistema para realizar sus actualizaciones en módulos posteriores.

3.3.6 Desarrollo de la tercera iteración

Para desarrollar el módulo de agregar clientes se basó directamente en el documento denominado "Formato de información comercial" que es utilizado para almacenar los datos que necesita la empresa para desarrollar los proyectos. Este formulario es llenado principalmente por los vendedores, pero puede ser usado también por los demás niveles de usuarios dado que la empresa fomenta que sus empleados también puedan obtener ingresos extras al tener sus propios clientes convirtiéndose indirectamente en vendedores. Este formulario se muestra en la figura 11 que se presenta a continuación.

Información Comercial

Datos de la Empresa

Nombre:

RFC:

Ramo/Giro:

Dirección

Calle:

Número:

Interior:

Colonia:

Código Postal:

Ciudad:

Estado: ▼

Figura 11. Frammento del formulario para añadir cliente

Así mismo se creó la ventana de agregar proyecto que permitirá a los integrantes del área gestión de proyectos asignar un tipo de proyecto y una breve descripción.

3.3.7 Desarrollo de la cuarta iteración

La pantalla de editar cliente sólo es accesible por el creador del cliente a través de la pestaña “acerca de” donde es visible la información del cliente y en futuras iteraciones se agregará el historial de actividades en esta misma ventana como se muestra en la figura 12.



Figura 12. Pantalla de información de proyecto e historial de actividades

3.3.8 Desarrollo de la quinta iteración

La pantalla añadir evento contiene los siguientes parámetros:

- Escoger tipo de actividad
- Añadir una descripción/resumen de lo realizado
- Añadir un archivo (opcional)
- Escoger al usuario que deberá recibir el proyecto para la siguiente actividad

Adicionalmente existen actividades denominadas “otras” que hace referencia a cualquier otro tipo de actividad que no esté contemplada en la línea estándar del desarrollo de un proyecto pero que sea relevante o derivada de la misma línea, de forma que al leer el historial puedan saber todas las actividades que han transcurrido con respecto al proyecto. Este formulario se muestra en la figura 13 a continuación.

Figura 13. Formulario para agregar actividad

3.3.9 Desarrollo de la sexta iteración

Esta pantalla es compleja por el hecho de todo lo que se puede realizar algunos de los informes que se pueden generar (según el nivel de usuario es distinta la información que se puede obtener):

- Clientes registrados en un periodo de tiempo determinado
- Proyectos registrados en un periodo de tiempo determinado
- Proyectos terminados en un periodo de tiempo determinado
- Proyectos de un cliente específico
- Historial de actividades de un proyecto específico
- Historial de actividades en un periodo de tiempo determinado

Para poder definir qué tipo de informe se desea generar se hace juego con formularios dinámicos haciendo uso de php y JQuery. Además, el formulario puede descargarse en formato PDF o XLX según la elección del usuario. En la figura 14 podremos observar el formulario que se llena para generar un informe.

Figura 14. Formulario para generar informe

3.4 Pruebas

Las pruebas realizadas a las funcionalidades del sistema permiten constatar que la información dada en la recopilación de la información y las historias de usuario hayan sido completamente cumplidas. Por eso se diseñaron algunas pruebas de aceptación para verificar el cumplimiento de los requisitos. La prueba de aceptación de la tabla 20 a la 25 nos muestran el resultado del estudio realizado a cada una de las 6 iteraciones o versiones del sistema.

Prueba de aceptación 01			
Nombre	Gestión de usuarios		
Descripción	Evaluar el acceso y gestión de usuarios del sistema		
Elemento	Actividades	Estado	Observaciones
Pantalla de Inicio de sesión	Validación de credenciales	OK	
	Acceso al sistema	OK	
Administrar Usuarios	Añadir Usuario	OK	
	Modificar Usuario	OK	
	Eliminar Usuario	OK	

Tabla 20. Prueba de aceptación 01

Prueba de aceptación 02			
Nombre	Pantalla principal		
Descripción	Evaluar la información contenida en la pantalla principal		
Elemento	Actividades	Estado	Observaciones
Pantalla principal	Muestra nombre de usuario	OK	
	Muestra clientes según el nivel de usuario	OK	
	Muestra clientes con más de un proyecto	OK	
	Muestra clientes sin proyectos	OK	
	Las opciones de la barra de navegación cambian si es administrador o no	OK	

Tabla 21. Prueba de aceptación 02

Prueba de aceptación 03			
Nombre	Adición de información comercial		
Descripción	Evaluar el correcto funcionamiento de las pantallas donde se agrega la información inicial comercial		
Elemento	Actividades	Estado	Observaciones

Agregar Clientes	Agregar un cliente	OK	
	Validación de consistencia de datos	OK	
Agregar Proyecto	Agregar un proyecto a un cliente sin proyecto	OK	
	Agregar un proyecto a un cliente con proyectos existentes	OK	

Tabla 22. Prueba de aceptación 03

Prueba de aceptación 04			
Nombre	Actualización de clientes		
Descripción	Evaluar las interfaces que muestran y permiten modificar la información de los clientes		
Elemento	Actividades	Estado	Observaciones
Editar cliente	Actualizar la información del cliente	OK	
Información de cliente	Muestra información según el nivel de usuario	OK	
	Muestra botón para modificar cliente sólo al creador	OK	
	Muestra tabla de historial de actividades	OK	

Tabla 23. Prueba de aceptación 04

Prueba de aceptación 05			
Nombre	Añadir Eventos		
Descripción	Evaluar el formulario de añadir eventos		
Elemento	Actividades	Estado	Observaciones
Añadir evento	Muestra diferentes actividades según el nivel de usuario	OK	
	Permite añadir una descripción	OK	
	Permite añadir un archivo adjunto	OK	
	Permite escoger al encargado de la siguiente etapa	OK	
	Envía una notificación vía correo electrónico al encargado de la siguiente etapa	OK	
	Se agrega al	OK	

	historial de actividades de la pantalla información del cliente		
--	---	--	--

Tabla 24. Prueba de aceptación 05

Prueba de aceptación 06			
Nombre	Informes		
Descripción	Evaluar los distintos informes que puede generar el sistema		
Elemento	Actividades	Estado	Observaciones
Informes	Muestra opciones según el nivel de usuario	OK	
	Se puede escoger entre diversos informes	OK	
	Permite escoger periodos de tiempo	OK	
	Permite escoger el formato del archivo a descargar	OK	

Tabla 25. Prueba de aceptación 06

Gracias a las pruebas de funcionalidad realizadas podemos concluir que el sistema cumplió con los requerimientos establecidos por la empresa a través de las historias de usuario previamente presentadas dando además la posibilidad de realizar actualizaciones y mejoras según crezcan las necesidades de la empresa a futuro.

3.5 Lanzamiento

Para el lanzamiento provisional se hizo uso del servidor gratuito 000webhost como se mencionó previamente esto para realizar las pruebas de funcionamiento que se pudieran realizar los cambios pertinentes, gracias a esto el proyecto está listo para subirse al servidor de la empresa en cuanto la aprobación de la mesa directiva esté hecha.

4. Conclusiones y recomendaciones

Durante el desarrollo de este proyecto pude reconocer la importancia de tener constante y frecuente comunicación con los usuarios finales del proyecto puesto que ellos nos pueden dar una vista menos técnica y más real del funcionamiento del sistema.

Me sentí bastante presionado en cuanto a tiempos puesto que es un proyecto de 600 horas en relativamente pocos meses entonces me gustaría que se pudiera gestionar mejor éstos. Inclusive debido a la pandemia y la cuarentena pese a que lo lógico sería que tuviéramos más tiempo para poder realizar cómodamente nuestro proyecto la realidad es que tenemos muchas más cosas que hacer y presión por lo cual tanto el rendimiento como el tiempo se ven severamente reducido. Creo que si hubiese estado en una estadía en una situación diferente el resultado y el proceso hubieran sido más cómodos y con mejor rendimiento de trabajo.

También siento que es importante para futuros trabajos buscar siempre el tener equipos de trabajo puesto que, aunque en las estadías es preferente que sea un proyecto individual ciertamente es muchísimo mejor el trabajar con un equipo dado que las tareas, responsabilidades son divididas y también da la oportunidad de tener retroalimentación entre los mismos integrantes del equipo de trabajo.

5. Anexos

Apéndice A

```
<?php
session_start();

if(!$_SESSION['ok'] || !isset($_SESSION['ok'])){
    header("Location:index.php");
}

?>

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <!-- CSS -->

    <link rel="stylesheet"
href="https://cdn.jsdelivr.net/npm/bootstrap@4.5.3/dist/css/bootstrap.min.css"
integrity="sha384-TX8t27EcRE3e/ihU7zmQxVncDAY5uIKz4rEkGIXeMed4M0jlfIDPvg6uqKI2xXr2"
crossorigin="anonymous">

    <!-- jQuery and JS bundle w/ Popper.js -->

    <script src="https://code.jquery.com/jquery-3.5.1.slim.min.js" integrity="sha384-
DfxDz2htPH0lsSSs5nCTpuj/zy4C+OGpamoFVy38MVBnE+IbbVYUew+OrCXaRkfj"
crossorigin="anonymous"></script>

    <script
src="https://cdn.jsdelivr.net/npm/bootstrap@4.5.3/dist/js/bootstrap.bundle.min.js"
integrity="sha384-ho+j7jyWK8fNQe+A12Hb8AhRq26LrZ/JpcUGGOn+Y7RsweNrtN/tE3MoK7ZeZDyx"
crossorigin="anonymous"></script>

    <title>Sistema de Gestión de Proyectos - REDCOMM</title>

    <style>

        body{

            background-color: lightgray;

        }

    </style>

</head>
```

Apéndice B

```
<?php

    $databaseHost = 'localhost';

    $databaseName = '*****';
```

```
$databaseUsername = '*****';  
  
$databasePassword = '*****';  
  
try {  
  
    $dbConn      =      new      PDO("mysql:host={$databaseHost};dbname={$databaseName}",  
$databaseUsername,  
  
    $databasePassword);  
  
    $dbConn->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);  
  
} catch(PDOException $e) {  
  
    echo $e->getMessage();  
  
    }  
  
?>
```

6. Referencias bibliográficas

- [1] Bernal Torres, 2006. Metodología de la investigación. México: Pearson/Educación. Fecha de Consulta: 4/noviembre/2020
- [2] URL: <https://devcode.la/blog/que-es-html> ¿Qué es HTML? Fecha de Consulta: 23/octubre/2020
- [3] URL: <https://www.php.net/manual/es/intro-what-is.php> ¿Qué es PHP? Página oficial de PHP. Fecha de Consulta: 23/octubre/2020
- [4] Eguiluz, J. Introducción a JavaScript. Uniwebsidad.com. Disponible en: <https://uniwebsidad.com/libros/javascript> Fecha de Consulta: 4/noviembre/2020
- [5] Eguíluz Javier, 2020. Introducción a CSS. Uniwebsidad.com. Disponible en: <https://uniwebsidad.com/libros/css>. Fecha de Consulta: 23/octubre/2020
- [6] Otto, M. y Thornton, J., 2020. Bootstrap 3, El Manual Oficial. Uniwebsidad.com. Disponible en: <https://uniwebsidad.com/libros/bootstrap-3> Fecha de Consulta: 23/octubre/2020
- [7] Berni Millet, P. y Gil de la Iglesia, D., 2010. Laboratorio De PHP Y MySQL. Barcelona: UOC Universidad Oberta de Catalunya. Fecha de Consulta: 4/noviembre/2020



Universidad Politécnica de Puebla
Ingeniería en Informática

Josué Ammón Ortiz Gallegos
Adrián A. Cienfuegos Pérez
Rebeca Rodríguez Huesca

Este documento se distribuye para los términos de la
Licencia 2.5 Creative Commons (CC-BC-NC-ND 2.5 MX)