



UNIVERSIDAD DE QUINTANA ROO
DIVISIÓN DE CIENCIAS E INGENIERÍA

EXPERIENCIA PROFESIONAL DE UN
INGENIERO EN REDES EN EL
DESARROLLO DE SOFTWARE

TRABAJO MONOGRÁFICO
PARA OBTENER EL GRADO DE

INGENIERO EN REDES

PRESENTA
DAVID MÁRQUEZ OJEDA

SUPERVISORES

MTI. MELISSA BLANQUETO ESTRADA

MSI. RUBÉN ENRIQUE GONZÁLEZ ELIXAVIDE

MTI. VLADIMIR VENIAMIN CABAÑAS VICTORIA



CHETUMAL QUINTANA ROO, MÉXICO, ENERO DE 2019



UNIVERSIDAD DE QUINTANA ROO

DIVISIÓN DE CIENCIAS E INGENIERÍA

TRABAJO MONOGRÁFICO TITULADO

EXPERIENCIA PROFESIONAL DE UN INGENIERO EN REDES EN EL DESARROLLO DE SOFTWARE

ELABORADO POR
DAVID MÁRQUEZ OJEDA

BAJO SUPERVISIÓN DEL COMITÉ DE SUPERVISIÓN Y APROBADO COMO
REQUISITO PARCIAL PARA OBTENER EL GRADO DE
INGENIERO EN REDES

COMITÉ DE SUPERVISIÓN

SUPERVISORA:


M.T.I. MELISSA BLANQUETO ESTRADA

SUPERVISOR:


M.S.I. RUBÉN ENRIQUE GONZÁLEZ ELIXAVIDE

SUPERVISOR:


M.T.I. VLADIMIR VENIAMIN CABAÑAS VICTORIA



CHETUMAL, QUINTANA ROO, MÉXICO, ENERO DE 2019

Resumen

La iniciativa Grupo RED ubicada en la ciudad de Chetumal, Quintana Roo realiza el reclutamiento y capacitación en desarrollo de software de profesionistas en su mayoría recién egresados en carreras afines a Tecnologías de la Información y Comunicación, con el objetivo de formarlos profesionalmente en JAVA y .Net.

Iniciando con el aprendizaje de los conceptos de OOP y patrones de diseño, a continuación, realizan prácticas que desarrollan por sí mismos para comprender los conceptos estudiados, una vez evaluados satisfactoriamente son asignados a un equipo para apoyar en tareas referentes al Frontend (HTML, CSS, JQuery y Javascript) y por último son incorporados a proyectos reales para experimentar el ambiente profesional que envuelve el desarrollo de software.

La metodología ágil que se emplea para la administración de los proyectos es Scrum, la cual permite finalizar los proyectos en el tiempo requerido y en el presupuesto establecido. Consiste en eliminar todos los procesos que no se encuentran directamente relacionados con la entrega de un producto o prototipo funcional y el cliente es un miembro activo del equipo de desarrollo y permite una comunicación constante entre el equipo y cliente con un alto grado de satisfacción del proyecto, al estar recibiendo módulos completos funcionales en vez de informes de avances teóricos que no representan la realidad del proyecto.

Agradecimientos

A la Universidad de Quintana Roo y en especial a los profesores de Ingeniería, quienes aportaron sus conocimientos para enriquecer los míos y siempre motivaron a buscar más allá de lo impartido en las aulas.

Dedicatoria

A mi esposa Seidi. Por con su paciencia, apoyo y motivación en los años que dedique para alcanzar este objetivo profesional.

A mi hija Citlaly. Por estar siempre apoyando a pesar de los periodos de ausencia.

Contenido

Agradecimientos	ii
Dedicatoria.....	iii
Capítulo 1. Introducción	3
Objetivos	5
Alcances y restricciones	6
Capítulo 2. Marco Contextual.....	7
Razón social y giro de la empresa	7
Producto	7
Antecedentes	8
Ubicación	8
Mercado.....	8
Ventas.....	9
Producción.....	9
Organización	10
Infraestructura	10
Local	10
Capítulo 3. Procedimiento de capacitación en el desarrollo de software	12
Plan de capacitación	12
Fase 1.....	12
Fase 2.....	13
Fase 3.....	13
Fase 4.....	14
Proyectos realizados con participación de los jóvenes	15
Capítulo 4. Enfoque de la empresa para el desarrollo de software	16
Capítulo 5. Rol del administrador del proyecto durante el proceso de desarrollo de software	21
Capítulo 6. Problemas más comunes en los equipos de desarrollo de software y las soluciones propuestas.....	29
Capítulo 7 Conclusiones	34

Bibliografía.....	vii
Glosario	viii
Abreviaturas.....	xi

Índice de Figuras y Tablas

Figura 1 Proceso de desarrollo de Software	8
Figura 2 Organización de la empresa Grupo RED	10
Figura 3 Plano edificio Grupo RED	11
Figura 4 Ciclo de vida del proyecto (Gido & Clements (2012)).....	24

Capítulo 1. Introducción

De acuerdo con el sitio web apiumhub.com (2018) en la página publicada: Estudio sobre la Situación actual del Software 2018: El desarrollo de software es un campo cuya constante es el cambio, todos los años aparecen nuevos lenguajes, arquitecturas de software, contenedores, metodologías, etc. Lo que hoy es lo más utilizado sea lenguaje o herramienta podría estar obsoleto para el siguiente año.

Las empresas de tecnología de México y el mundo tienen un desafío principal relacionado con la capacidad de entregar software que funcione mientras las características requeridas están al límite y, el segundo desafío es atraer y retener el talento, lo cual es un gran problema para las empresas de desarrollo.

Los equipos para superar el desafío de capacidad implementen soluciones como: optimizar los flujos de trabajo, priorización y aumento del número de personas participando en los proyectos, aplicación de métodos ágiles.

Para resolver el problema de talento los equipos contratan un proceso de captación de talento interno o externo para contactar a los candidatos. También realizan lanzamientos de programas de prácticas. Existen 4 áreas importantes en las que las compañías tecnológicas se enfocan: predisposición para aprender, encajar en el tipo de cultura de la empresa, experiencia laboral y evaluación de habilidades técnicas.

En cuanto a la retención de talento, el desafío en el trabajo y la cultura de equipo son las formas más populares para motivar a los desarrolladores a quedarse en las empresas y evitar la fuga de talentos.

Para medir el rendimiento de los desarrolladores en los equipos de software, los líderes de proyecto se basan en los factores: software que funciona, tareas completadas y legibilidad del código.

La principal causa de fracaso para la mayor parte de los líderes de proyecto son las expectativas poco realistas, seguidas de la estimación y falta de resultados claramente definidos.

El desarrollo profesional en el área de Tecnologías de la Información enfocado al desarrollo de software, desde diciembre de 2014 en la incubación de la empresa que actualmente es una realidad y se denomina Grupo RED, ha significado un reto importante para brindar soluciones Web a la medida de los clientes que requieren nuestros servicios.

Sin embargo, el complejo proceso que se realiza al integrar y administrar los equipos de desarrollo en un proyecto de software, que en muchos de los casos genera quejas en el personal, retrasos en la entrega, gastos extra y resultados deficientes generados por la falta de capacitación de los desarrolladores y la falta de comunicación con el cliente, me ha llevado a documentar este proceso para facilitar en gran medida la incursión de los jóvenes en el proceso de desarrollo de software y que nos permita reducir en gran medida la curva de aprendizaje de nuevas tecnologías.

Objetivos

General

Documentar los procedimientos de la integración y funcionamiento de los equipos de desarrollo de software para la empresa Grupo RED.

Específicos

- Documentar el procedimiento de capacitación en el desarrollo de software.
- Documentar el enfoque de la empresa Grupo RED para el desarrollo de software.
- Documentar el rol del administrador del proyecto durante el proceso de desarrollo de software.
- Documentar algunos de los problemas más comunes en los equipos de desarrollo de software y las soluciones propuestas.

Justificación

Con el vertiginoso avance tecnológico de la actualidad los clientes demandan soluciones que se ajusten a sus requerimientos futuros, volviéndose preponderante cubrir la necesidad de integrar equipos de desarrollo de software de alto rendimiento que finalicen el proyecto en tiempo, costo y completa satisfacción del cliente. Esto requiere de una minuciosa planeación que inicia desde las entrevistas para detectar talento, capacitación en nuevas tecnologías y metodologías ágiles de gestión, hasta el desarrollo en equipo como un solo ente para conseguir el objetivo final.

Convirtiéndose en una verdadera necesidad documentar las diferentes etapas del proceso de desarrollo de software para facilitar la integración de nuevos desarrolladores a las tecnologías emergentes.

Alcances y restricciones

La memoria de experiencia profesional comprende de enero de 2015 a diciembre de 2017.

Se define como alcance la documentación de los procedimientos de integración y funcionamiento de los equipos de desarrollo de software.

Capítulo 2. Marco Contextual

Razón social y giro de la empresa

La empresa donde se realiza la experiencia profesional se denomina “Grupo RED”, los productos que desarrolla principalmente son páginas web y aplicaciones móviles.

Tiene una antigüedad de 3 años en el mercado, inició como una empresa entre amigos desarrollando proyectos de software pequeños para empresas de la ciudad de Monterrey; actualmente cuenta con 8 jóvenes que concluyeron su capacitación y están laborando en empresas nacionales, 10 colaborando en un proyecto conjunto con una empresa de Monterrey y 2 iniciando su capacitación en el área de software.

Producto

El producto principal que se desarrolla en la empresa son páginas web dinámicas utilizando las tecnologías .Net 4, C#, ASP .Net MVC con Razor, SQL Server, Entity Framework Code First, Ninject DI, Bootstrap, Web API y AngularJS.

También se utilizan tecnología Java, Spring MVC 4, Spring Security, Servicios REST, MyBatis, MySQL | Oracle 10g+ | MSSQL Server 2008 R2+, Bootstrap, JQuery | Angular y Jqgrid Plugin.q

En el área de tecnologías para móviles se utiliza: Xamarin, Xamarin.Forms, XAML y UWP.

El proceso de desarrollo de software tiene la siguiente estructura:

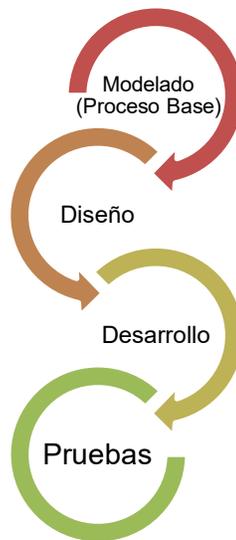


Figura 1 Proceso de desarrollo de Software

Antecedentes

Ubicación

La empresa se encuentra ubicada en la ciudad de Chetumal, municipio Othón P. Blanco del estado de Quintana Roo.

Mercado

Los productos están dirigidos a clientes de Monterrey y Guadalajara dentro del país; en Estados Unidos de América en las ciudades de Chicago, Austin y Dakota del Sur; actualmente está por iniciar un proyecto para una compañía de la India.

Ventas

El costo de los productos varía de acuerdo con el tiempo de la duración del proyecto en horas, número y tipo de recursos empleados.

Los ingresos mensuales varían, facturando un promedio de \$30,000.00 mensuales y venta anual de \$360,000.00 en 2016 y 2017.

Este año 2018 se estima un aumento de ingresos del 200% al 300%.

Producción

Se cuenta con una producción promedio de un proyecto de 336 horas cada 2 meses utilizando dos recursos al 100%, los cuales son un desarrollador Senior y un desarrollador Junior.

La productividad de desarrollo se estima basado en la experiencia de los desarrolladores en la plataforma tecnológica que se empleará para realizar el proyecto y para la administración se utiliza la metodología ágil denominada Scrum, la cual describimos con detalle en el capítulo 4.

Para asegurar la calidad del producto en conjunto con la calidad del proceso, al ser actualmente aspectos muy importantes en el desarrollo de software, se aplica la familia de normas ISO/IEC 25000, las cuales proporcionan una guía para el uso de la nueva serie de estándares internacionales llamada Requisitos y Evaluación de Calidad de Productos de Software (SQuARE - System and Software Quality Requirements and Evaluation).

ISO/IEC 25000 constituye una serie de normas basadas en ISO/IEC 9126 y en ISO/IEC 14598 cuyo objetivo principal es guiar el desarrollo de los productos de software mediante la especificación de requisitos y evaluación de características de calidad.

Organización

La organización de la empresa Grupo RED se observa en la Figura 2.

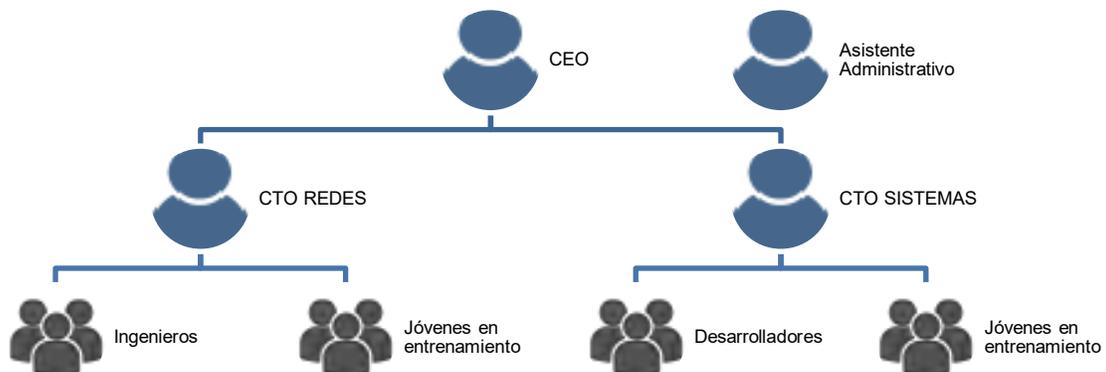


Figura 2 Organización de la empresa Grupo RED

Infraestructura

Local

La propiedad cuenta con 160 m² de terreno y 120m² de construcción dividido en 4 oficinas de 4m x 3m, cocina y baño.

Actualmente trabajan 16 personas en estas oficinas, distribuidos de acuerdo con la asignación de proyectos o capacitación.

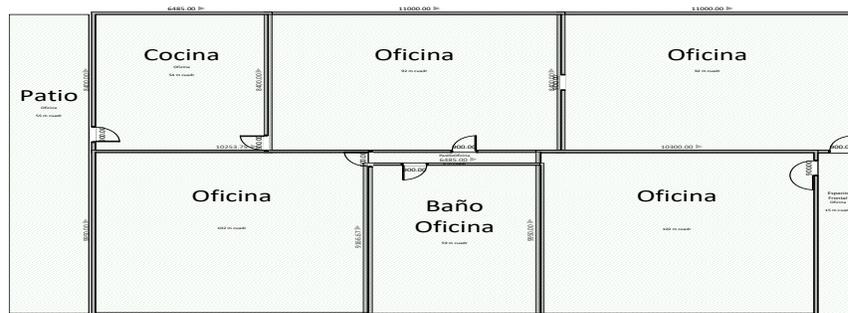


Figura 3 Plano edificio Grupo RED

Capítulo 3. Procedimiento de capacitación en el desarrollo de software

Plan de capacitación

El plan de capacitación para los jóvenes que se incorporan a Grupo RED presenta 2 vertientes de acuerdo al interés profesional de cada uno de ellos:

- .Net
- Java

Fase 1

La primera fase inicia con la capacitación básica que está formada por los siguientes conceptos:

Bases de la Programación Orientada a Objetos

- Herencia
- Polimorfismo
- Clases abstractas
- Interfaces
- Propiedades
- Métodos
- Encapsulamiento
- Diferencias entre clase abstracta e interfaz
- Clase concreta
- Composición

Conceptos de estudio para patrones de diseño

- Clasificación de patrones de diseño
- Nombres de los patrones de diseño
- Aplicación de los patrones de diseño.

Fase 2

En una segunda etapa aplican los conceptos en prácticas que desarrollan por sí mismos para comprender los conceptos estudiados.

Estas prácticas se desarrollan de acuerdo con la vertiente elegida en Visual Studio o Netbeans y se evalúan los conceptos aplicados y su funcionalidad.

Fase 3

En la tercera etapa, cambia el escenario de capacitación ahora se enfocan en desarrollo Web Front-end, aprendiendo los conceptos y practicando lo referente a:

- HTML
- CSS
- JQuery
- Javascript

Para conseguir los mejores resultados nos apoyamos en las herramientas en línea:

- Pluralsight¹ para cursos en vídeo.
- Safari² para consulta de libros.

En ambos casos contamos con suscripciones para que los jóvenes aumenten sus capacidades profesionales, correspondientes a las necesidades identificadas.

¹ <https://www.pluralsight.com/>

² <https://www.safaribooksonline.com/>

Fase 4

En la cuarta etapa, se incorporan a proyectos reales para experimentar el ambiente profesional que envuelve el desarrollo de software.

Las tareas consisten en realizar modificaciones al diseño (html y css), aplicar contenido dinámico de las páginas web (jQuery, Javascript), replicar código en distintos componentes y de acuerdo al tiempo de capacitación codificación de módulos completos del proyecto, utilizando las versiones más recientes de esta tecnologías.

Proyectos realizados con participación de los jóvenes

Esta inducción de los jóvenes al desarrollo profesional de software permite que exploten sus habilidades al incrementar la complejidad de las tareas a realizar conforme avanza el tiempo de capacitación y al mismo tiempo obtienen experiencia al participar en un entorno laboral real con proyectos planeados a ejecutarse en tiempos definidos.

Proyectos profesionales realizados con la participación de los jóvenes:

Proyectos profesionales	Tecnologías
Sitio web de Villas Club. Administración de una estancia, que cuenta con 10 centros en la ciudad de Monterrey, N.L.	.Net 4, C#, ASP .Net MVC with Razor, SQL Server 2014, Entity Framework Code First, Ninject DI, Bootstrap, Web API, AngularJS.
Sitio web de Valkiria EVO. Administración del punto de venta de restaurantes, el cual cuenta con clientes en distintas ciudades de país.	.Net 4, C#, ASP.Net Web Forms, SQL Server 2014, Dapper ORM, Bootstrap, Telerik Web Reporting.
Sitio web Weir Minerals. Administración y seguimiento de pedidos de productos de distintas partes del mundo.	.Net 4, C#, ASP.Net MVC, SQL Server 2014, Bootstrap, JavaScript, HTML, CSS, JQuery.
Sitio web de OpenService. Administración de tickets de soporte para las tiendas Oxxo de todo el país.	Net 4, C#, SQL Server 2014, Sencha ExtJs 6, JavaScript, HTML, CSS.
Calculadora ROI en sitio web de Linkaform, calcula el retorno de inversión de los clientes en su migración de formas físicas a digitales.	.Net 4, C#, ASP.Net Web API, SQL Server 2014, Bootstrap, JavaScript, HTML, CSS, JQuery.
Aplicación móvil de inventarios de infraestructura Pueblos Mágicos.	Net, C#, Xamarin, AXML.
Aplicación móvil para red social de mascotas DogMatch.	Net, C#, Xamarin, AXML.

Capítulo 4. Enfoque de la empresa para el desarrollo de software

El enfoque que se utiliza para el desarrollo de software es basado en metodología ágil denominada Scrum, que consiste en eliminar todos los procesos que no se encuentran directamente relacionados con la entrega de un producto o prototipo funcional y el cliente es un miembro activo del equipo de desarrollo.

En este método se establece un calendario de reuniones con el cliente, que pueden ser diarias, semanales o cada dos semanas y en cada reunión se entrega un avance funcional que puede probar y retroalimentar para realizar los cambios y ajustes requeridos con tiempos de respuesta cortos por parte del equipo de desarrollo.

En la reunión cada miembro del equipo presenta los requerimientos de información o reglas de negocio que obstaculizan el avance del proyecto, así como su programa de entregables.

Esto permite una comunicación constante entre el equipo y cliente con un alto grado de satisfacción del proyecto, al estar recibiendo módulos completos funcionales en vez de informes de avances teóricos que no representan la realidad del proyecto.

Descripción del proceso de un proyecto con Scrum Shuterland (2016):

1. Se elige un **responsable del proyecto**. Este individuo es quien posee la visión de lo que se va a desarrollar. Tomar en cuenta los riesgos y recompensas, qué es posible, qué puede hacerse, qué le apasiona.
2. Seleccionar un **equipo**. ¿Quiénes serán las personas que harán efectivamente el trabajo? Este equipo debe contar con todas las habilidades necesarias para tomar la visión de los responsables del proyecto y hacerla realidad. Los equipos deben ser pequeños de tres a nueve personas por regla general
3. Elegir un **Scrum Master**. Ésta es la persona que capacitará al resto del equipo en el enfoque de Scrum y que ayudará a eliminar al equipo de todo lo que atrasa.

4. Crear y priorizar una **bitácora del proyecto**. Se trata de una lista de alto nivel de todo lo que debe hacerse para volver realidad la visión. Esta bitácora existe y evoluciona durante el periodo de vida del proyecto; es la guía que permite lograr los objetivos. En un momento dado, la bitácora del proyecto es la visión definitiva de “todo lo que el equipo podría hacer, en orden de prioridad”. Hay sólo una bitácora del proyecto; esto significa que el responsable debe tomar decisiones de priorización en todo el espectro. El responsable del proyecto debe consultar tanto a todos los interesados como al equipo para cerciorarse de que representa lo que se necesita y lo que se puede hacer.

5. Mejorar y estimar la **bitácora del producto**. Es crucial que la gente que realmente se hará cargo de los elementos de la bitácora del producto estime cuanto esfuerzo implicarán. El equipo debe examinar cada elemento de la bitácora y ver, si en efecto, es viable. ¿Hay información suficiente para llevar a cabo el elemento? ¿Este es lo bastante pequeño para estimarse? ¿Existe una definición de “terminado”; es decir, todos están de acuerdo en los criterios que deben cumplirse para poder decir que algo está “terminado”? ¿Esto crea valor visible? Cada elemento debe poder mostrarse y demostrarse y (es de esperar) entregarse. No calcular la bitácora en horas, porque la gente es pésima para esto. Calcula por tamaño relativo: pequeño, mediano o grande. O, mejor todavía, usa la serie de Fibonacci y estima el valor puntual de cada elemento: 1, 2, 3, 5, 8, 13, 21, etcétera.

6. **Planeación del sprint**. Ésta es la primera de las reuniones de Scrum. El equipo, el Scrum Master y el responsable del proyecto se sientan a planear el sprint. Los sprints son siempre de extensión fija, inferior a un mes. La mayoría de la gente ejecuta en la actualidad de uno a dos sprint semanales. El equipo examina el inicio de la bitácora y pronostica cuanto puede llevar a cabo en ese sprint. Si el equipo ha pasado por varios sprints, debe considerar el número de puntos que acumulo en el más reciente. Este número se conoce como velocidad del equipo. El Scrum Master y el equipo deben tratar de aumentar ese número en cada sprint. Ésta es otra oportunidad para que el equipo y el responsable del proyecto

confirman que todos comprenden a la perfección como esos elementos cumplirán la visión. Durante esta reunión todos deben acordar asimismo una meta de sprint, que todos han de cumplir en este sprint.

Uno de los pilares de Scrum es que cada vez que el equipo se compromete con lo que cree que puede terminar en un sprint, eso se queda ahí. No puede cambiar ni crecer. El equipo debe ser capaz de trabajar en forma autónoma a lo largo del sprint para terminar lo que pronosticó que podía hacer.

7. **Vuelve visible el trabajo.** La forma más común de hacerlo en Scrum es crear una **tabla de Scrum** con tres columnas: Pendiente, En proceso y Terminado. Notas adhesivas representan los elementos por llevar a cabo y el equipo avanza por la tabla conforme los va concluyendo, uno por uno.

Otra manera de volver visible el trabajo es crear un diagrama de finalización. En un eje aparece el número de puntos que el equipo introdujo en el sprint y en el otro el número de días. Cada día el Scrum Master suma el número de puntos completados y los grafica en el diagrama de finalización. Idealmente, habrá una pendiente descendente que conduzca a cero puntos para el último día del sprint.

8. **Reunión diaria o Scrum diario.** Éste es el pulso del Scrum. Cada día, a la misma hora, durante no más de quince minutos, el equipo y el Scrum Master se reúnen y contestan tres preguntas:

¿Qué hiciste ayer para ayudar al equipo a terminar el sprint?

¿Qué harás hoy para ayudar al equipo a terminar el sprint?

¿Algún obstáculo te impide o impide al equipo cumplir la meta del sprint?

Eso es todo, en eso consiste la reunión. Si se prolonga más de quince minutos se está haciendo mal. Lo que esto hace es ayudar al equipo a saber exactamente dónde se encuentra todo en el curso de un sprint. ¿Todas las tareas serán terminadas a tiempo? ¿Hay oportunidades de ayudar a otros miembros del equipo a vencer obstáculos? Las tareas no se asignan desde arriba; el equipo es autónomo: él lo hace. Tampoco se rinden informes detallados a la dirección. El Scrum Master se encarga de eliminar los obstáculos, o impedimentos, contra el progreso del equipo.

9. **Revisión del sprint o demostración del sprint.** Ésta es la reunión en la que el equipo muestra lo que hizo durante el sprint. Todos pueden asistir, no solo el responsable del proyecto, el Scrum Master y el equipo, sino también los demás interesados, la dirección, clientes, quien sea. Ésta es una reunión abierta en la que el equipo hace una demostración de lo que se pudo llevar a terminado durante el sprint.

El equipo debe mostrar únicamente lo que satisface la definición de terminado, lo total y completamente concluido y que puede entregarse sin trabajo adicional. Esto puede no ser un producto terminado, pero sí una función concluida de uno de ellos.

10. **Retrospectiva del sprint.** Una vez que el equipo ha mostrado lo que logró en el sprint más reciente –la cosa “terminada” y en posibilidad de enviarse a los clientes en busca de retroalimentación–, pensar en qué marchó bien, qué pudo haber marchado mejor y qué puede mejorarse en el siguiente sprint. ¿Cuál es la mejora en el proceso que como equipo pueden implementar de inmediato?

Para ser eficaz, esta reunión requiere de cierto grado de madurez emocional y una atmosfera de confianza. La clave es que no se trata de buscar a quién culpar; lo que se juzga es el proceso. ¿Por qué tal cosa ocurrió de tal manera? ¿Por qué pasamos por alto tal otra? ¿Qué podríamos hacer más rápido? Es crucial que la gente, como equipo, asuma la responsabilidad de sus procesos y de sus resultados y busque soluciones también como equipo. Al mismo tiempo, debe tener fortaleza para tocar los temas que le incomodan de un modo orientado a la solución, no acusatorio. Y el resto del equipo ha de tener la madurez de oír la realimentación, aceptarla y buscar una solución, no ponerse a la defensiva.

Al final de la reunión el equipo y el Scrum Master deben acordar una mejora en el proceso que implementarán en el siguiente sprint. Esa mejora al proceso, también llamada kaizen, debe incorporarse en la bitácora del sprint siguiente, con pruebas de aceptación. De esta manera, el equipo podrá ver fácilmente si en verdad implementó la mejora y qué efecto tuvo ésta en la velocidad.

11. Comenzar de inmediato el ciclo del siguiente sprint, tomando en cuenta la experiencia del equipo con los impedimentos y mejoras del proceso.

Capítulo 5. Rol del administrador del proyecto durante el proceso de desarrollo de software

Los proyectos tecnológicos presentan un verdadero desafío en su realización, para lograr los resultados deseados se debe iniciar con una planeación completa del ciclo de vida, tomando en cuenta todos los aspectos que enmarcan su contexto y aplicar un método eficaz para evitar retrasos o aumentos de presupuesto durante el desarrollo y concluirlos en tiempo superando los criterios de aceptación establecidos, así como las expectativas del cliente para lograr un cierre satisfactorio.

Gido & Clements (2012) definen un proyecto como “un esfuerzo para lograr un objetivo específico por medio de una serie particular de tareas interrelacionadas y el uso eficaz de los recursos”.

Frame (2000) menciona que el jefe de proyecto ideal presenta las características siguientes:

- Comprensión completa de los objetivos.
- Comprender las necesidades de los miembros del equipo.
- Prestar atención a los detalles.
- Comprometido con el proyecto y estar dispuesto a dedicarle largas horas.
- Capacidad para enfrentarse a los fracasos y desilusiones.
- Capacidad de negociación para obtener recursos.
- Ser práctico y orientado a obtener resultados.
- Ser consciente de los costos y tener capacidades empresariales básicas.
- Ser político para saber lo que hay que hacer y lo que no se debe hacer.
- Tener alta tolerancia a la ambigüedad.

Y describe los pasos para una gestión de proyectos:

- Tratamiento de la complejidad, es uno de los grandes desafíos a los que se enfrentan las organizaciones y los individuos, por lo que una rigurosa disciplina de proyecto con la creación y actualización de métodos, el estudio detallado de

los datos de programación, costos y la aplicación de fórmulas para el cambio es trabajo duro que nos lleva a aprender a manejar la complejidad.

- El cambio, es inevitable, por lo que se debe aceptar y desarrollar estrategias para manejarlo, éstas caen en dos categorías, la primera es seguir la corriente y reconocer que el cambio puede ser constructivo para responder necesidades y requisitos de los clientes, la siguiente es resistirse al cambio ya que no todo cambio es bueno, por lo que los cambios antojadizos son situaciones caprichosas del estado de ánimo y nos llevan al incumplimiento de los costes y plazos. De esta manera se puede decir que la dirección eficiente de proyectos es aquella que pueda distinguir entre el cambio bueno o malo y que esté preparado para controlar el cambio para sacar el mejor provecho posible.
- Riesgo, es una preocupación generalizada de las empresas y la administración pública, ya que los avances tecnológicos llegan con los ciudadanos comunes con información masiva de datos que antes eran inaccesibles, pero cuando se usan de manera efectiva nos ayudan a tomar decisiones fundamentadas y de esta manera se reduce el riesgo.

Satisfacción del cliente, esto implica que las organizaciones de proyectos abandonen sus viejos paradigmas de cuestiones técnicas y que se enfoquen a los nuevos que se centran en los clientes, aunque los proyectos deben responder a las necesidades muchas veces conflictivas de múltiples clientes.

Cleland and King (1990) señalan que un proyecto puede verse en tres etapas:

- Formación y dirección, los objetivos deben identificarse y entenderse con claridad, participan la alta dirección y el Administrador de Proyectos.
- Ejecución, puede haber correcciones de media carrera originadas por la administración del proyecto.
- Consumación, se estiman los resultados y se reconoce a los miembros del equipo.

También mencionan que las complejidades en los problemas de las organizaciones se incrementarán en el futuro, por lo que conllevará a más dificultad y necesitará una alta dirección ya que en la Administración de Proyectos recae la autoridad, responsabilidad, planeación, controlar y la resolución de conflictos, así que manejar bien los problemas de autoridad, ser diestro en planear, comunicar y manejar conflictos, influyen en el éxito probable de un proyecto.

Chamoun (2002) señala que, en la administración tradicional, desarrollamos un plan considerando sólo tiempo y costo, en vez de las nueve áreas de la APP (Administración Profesional de Proyectos) lo que causa que la comparación no sea real o confiable, al inicio se reportan buenos avances, pero en medio y al final existen desfases considerables que resultan en proyectos retrasados y fuera de presupuesto y no se utiliza el tiempo de corrección por falta de veracidad en la información.

Nos menciona que las fases en todo proyecto son las siguientes:

- Optimismo general, se inicia con esperanza de algo nuevo y dan buenos augurios.
- Desorientación, no se sabe por dónde comenzar, existen muchas ganas, se inicia con lo que se considera más urgente, pero se posterga lo importante.
- Período de desorden incontrolado, no todos los proveedores cumplen como dijeron, el alcance sigue sin definirse y cambia constantemente.
- Alarma y caos, no se cumple la fecha de entrega, se hacen pagos no considerados, el presupuesto se agota y no se ha logrado ni el 70% de avance, existen problemas con la calidad.
- Identificar al culpable.
- Las amenazas, intrigas, acusaciones y desesperación.
- Castigo, se despiden a los proveedores y a personal que menos culpa tienen.
- Recuperación del optimismo, ya eliminados los malos, inicia el segundo aliento.
- Terminación a como dé lugar, se trabaja a presión, tiempo extra, fines de semana, promesas y más presión.
- Condecoración y premios a los no participantes, el personal desarrollador del proyecto se encuentra cansado y terminando los últimos detalles y quienes no

participaron están frescos para adjudicarse el mérito, levantarse el cuello y recibir elogios.

Por otra parte, Gido & Clements (2012) establecen el ciclo de vida de un proyecto en cuatro fases: inicio, planeación, ejecución y cierre (Figura 1) y la duración de este ciclo es variable desde semanas o años dependiendo del contenido, complejidad y magnitud de los proyectos.

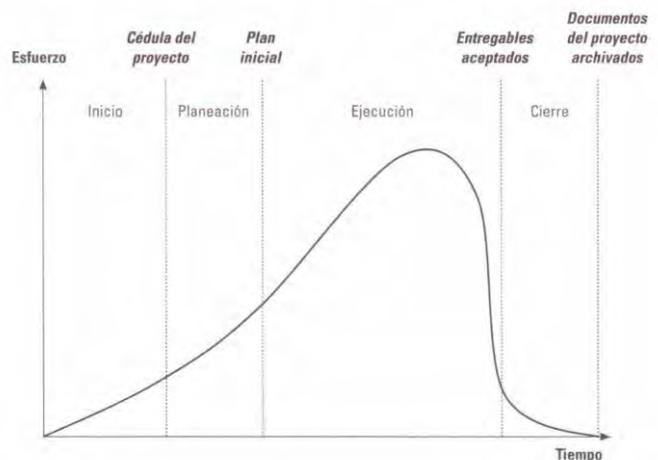


Figura 4 Ciclo de vida del proyecto (Gido & Clements (2012))

En Grupo RED, el rol del PM está basado en metodología Scrum y consiste en:

- Organizar todas las tareas del proyecto en elementos pequeños y manejables que puedan estimarse.
- Establecer la definición de “terminado” para cada tarea.
- Incorporar todas las reglas de negocio, requisitos legales, aseguramiento de la calidad y los reportes de procesos que los equipos deben cumplir.
- Convertir cada tarea a realizar en una historia corta que defina al personaje, usuario o cliente: el individuo que usará lo que se va a producir.
- Enumerar el trabajo en orden de prioridad, ponderando lo más valioso para el proyecto para realizarlo primero.
- Planear en detalle lo suficiente para cumplir el próximo sprint y calcular el resto del proyecto en líneas generales. Ir afinando el plan a lo largo del proyecto, no dejarlo terminado desde el principio.

- Determinar cuánto tiempo y esfuerzo implicará el proyecto, para esto se asigna un valor a cada tarea de acuerdo a su complejidad basado en los números de la serie Fibonacci, ya que están lo bastante separados entre sí para advertir fácilmente su diferencia.
- Organizar las reuniones de planeación del sprint. Estas congregan a todos los participantes, quienes analizan la lista de historias a ejecutar y determinan cuales podrán terminarse al final para mostrarse al cliente y comprobar que poseen valor real.
- Conocer la velocidad del equipo. Conociendo esta velocidad se puede contestar al fin la pregunta acerca de cuándo estará terminado el trabajo, porque sabe cómo medir lo que hace el equipo.

Dirigir la reunión diaria del equipo para eliminar los impedimentos que evitan que la velocidad aumente y alcance su nivel óptimo.

Ejemplo de Historias de Proyecto web para gestión de Seriales alfanuméricos:

Project Web para gestión de Seriales alfanuméricos de Punto de Venta de Restaurantes

	Name	Effort (hrs)
US 01	User Story 1 Diseñar web para gestionar los seriales del PDV	40
US 02	User Story 2 Creación de Base de datos y conexión a sitio web	24
US 03	User Story 3 Generación de Controlador de Sucursales	40
US 04	User Story 4 Generación de Seriales alfanuméricos de 25 caracteres	40
	Sum	144

Tabla 1 Resumen del Proyecto

US 01	User Story 1 Diseñar web para gestionar los seriales del PDV
--------------	---

Description	Para facilitar la administración y agilizar la respuesta a los requerimientos de los clientes que realizan sus pagos utilizaremos una página web, la cual permitirá acceder desde cualquier lugar y dispositivo.
Acceptance Criteria	<ul style="list-style-type: none"> * Diseño responsivo para cualquier tipo de dispositivo. * Mostrara en la parte superior un cuadro de búsqueda para filtrar la lista de Sucursales y botón para agregar Nueva sucursal. * Presentara la lista de Sucursales conteniendo: Núm. De Sucursal, Tiempo restante para vencimiento del serial en días, Fecha de corte, Serial activo, Días de vigencia para aplicar al nuevo serial, Cuadro de selección para aplicar Multipagos y Multicortes, Botón de corte de servicio. * Los Días de vigencia visualizaran un semáforo de acuerdo a los días restantes: más de 5 verde, 0 a 5 amarillo, menos de 0 rojo.
Effort (hrs)	40

Subtasks	Description	Effort (hrs)
US 01 - 1	Creación de estructura HTML de sitio web responsivo	8
US 01 - 2	Insertar elementos de formulario a sitio web	4
US 01 - 3	Implementación de Bootstrap a estructura HTML	12
US 01 - 4	Aplicación de estilos basados en CSS a sitio web	16

Tabla 2 Historia de Usuario 1

US 01	User Story 2 Creación de Base de datos y conexión a sitio web
--------------	--

Description	El sitio web almacenará la información que se gestiona en el sitio web mediante una Base de Datos que permita el acceso fácil y rápido para responder con prontitud a los requerimientos de los clientes.
Acceptance Criteria	<ul style="list-style-type: none"> * Contendrá una Base de datos normalizada con tablas para Sucursales y Seriales. * Tabla Sucursales con las Columnas: ID, Nombre, EmpresaID, SucursalID, DiaDeCorte, FechaDeCorte, IdentificadorDeEquipo, VersionPDV, FechaUltimaActualizacion, Activo. * Tabla Seriales con las Columnas: ID, SucursalID, Serial, FechaInicio, FechaFin, Consumido, FechaDescarga, Activo. * Creación del Proyecto ASP .Net MVC with Razor en Visual Studio 2015. * La conexión de la DB al Proyecto será mediante Entity Framework Code First. * Las versiones del Proyecto se gestionarán mediante Github.
Effort (hrs)	24

Subtasks	Description	Effort (hrs)
US 02- 1	Creación de Base de Datos y Tablas	6
US 02- 2	Creación de Proyecto ASP .Net MVC with Razor en Visual Studio 2015	3
US 02- 3	Realizar la conexión de la Base de Datos al Proyecto mediante EF Code First	3
US 02- 4	Crear los Modelos requeridos para la conexión a la Base de Datos.	4
US 02- 5	Configurar Github como versionador del Proyecto	8

Tabla 3 Historia de Usuario 2

US 03	User Story 3 Generación de Controlador de Sucursales
--------------	---

Description	Para gestionar la información de las Sucursales y Seriales que gestionara el sitio web, se requiere un controlador con los métodos necesarios para realizar esta tarea.
Acceptance Criteria	<ul style="list-style-type: none"> * Contendrá los siguientes métodos: Index, Create, Edit, Multipagos, Pagar, Corte. * Método Index presentará la información de Sucursales y Seriales filtrada por el cuadro de búsqueda y ordenada alfabéticamente por Nombre de Sucursal. * Método Create insertará una nueva Sucursal a la DB. * Método Edit modificará la información de la Sucursal seleccionada. * Método Multipagos generará Seriales validos por el tiempo especificado en cada una de las Sucursales seleccionadas. * Método Pagar genera un serial valido por un mes a la Sucursal actual. * Método Corte crea un serial invalido para cancelar el Servicio a la Sucursal actual.
Effort (hrs)	40

Subtasks	Description	Effort (hrs)
US 03 - 1	Crear Controlador Sucursales	1
US 03 - 2	Implementar método Index	9
US 03 - 3	Implementar método Create	4
US 03 - 4	Implementar método Edit	4
US 03 - 5	Implementar método Multipagos	10
US 03 - 6	Implementar método Pagar	8
US 03 - 7	Implementar método Corte	4

Tabla 4 Historia de Usuario 3

US 04	User Story 4 Generación de Seriales alfanuméricos de 25 caracteres
--------------	---

Description	Requerimos proteger el Punto de Venta de Restaurantes de copias ilegales mediante un mecanismo de Seriales que puedan ser generados en línea mediante una página web.
Acceptance Criteria	<ul style="list-style-type: none"> * Longitud de 25 caracteres alfanuméricos. * Generados de forma automática con un mes de validez. * Puede especificarse el número de días de validez, en este caso sustituye el mes de validez por la cantidad de días especificada. * El periodo de días inicia desde 1 día después de la fecha de corte. * En caso de la fecha actual sea mayor a 10 días después del corte, la fecha de corte toma el valor de la fecha actual.
Effort (hrs)	40

Subtasks	Description	Effort (hrs)
US 04 - 1	Crear clase con sus respectivos métodos para Generar Seriales alfanuméricos de 25 caracteres	24
US 04 - 2	Crear método Seriales en Controlador Sucursales para recibir lista de pagos y devolver Seriales correspondientes	16

Tabla 5 Historia de Usuario 5

Capítulo 6. Problemas más comunes en los equipos de desarrollo de software y las soluciones propuestas

Iniciarse en la industria del desarrollo de software como desarrollador representa un reto a menudo difícil de enfrentar, sobre todo cuando somos novatos en el área. Los errores y la frustración son la constante, por lo que el deseo de abandonar permanece latente. Sin embargo, pese a todos los malos augurios, la perseverancia y capacitación apoyan para mejorar cada día nuestras capacidades técnicas, conocimientos y actitudes, reduciendo el tiempo para superar la curva de aprendizaje.

La interacción con los equipos locales o remotos de los que formamos parte es otro tema importante para tener en cuenta, a menudo la falta de comunicación representa un problema que se convierte en grave y ocasiona una separación del desarrollador del equipo. Por esta razón, debemos mantener abierto los canales de comunicación con todos los miembros del equipo y apoyarlos para finalizar las historias de cada sprint con éxito.

Los problemas que se presentaron en mi experiencia como desarrollador en los distintos equipos locales y remotos de los cuales forme parte, así como la forma de solucionarlos los describo a continuación:

1. Errores de sintaxis.

Un error muy común al escribir código es escribir incompletas las sentencias o instrucciones, cuantas veces olvidamos finalizar con el punto y coma, cerrar los paréntesis de forma adecuada y teclear demás o de menos las sentencias, variables o constantes que manejamos en el código fuente de nuestro proyecto.

Solución: Con el avance de la tecnología es relativamente fácil corregir este tipo de errores, ya que la mayoría de los IDEs actuales proporcionan herramientas para revisar el código en tiempo real, visualizando la sintaxis completa o señalando la falta de un elemento mientras escribimos. Únicamente debemos

prestar atención a las advertencias y nos ahorraremos algunos dolores de cabeza.

2. Depurar correctamente

En ocasiones el código parecía estar correctamente escrito y el proyecto compilaba sin errores. Sin embargo, los resultados no eran los esperados o simplemente no aparecen en la pantalla. Después de investigar en internet y preguntar a los compañeros comprendí que el problema radicaba en la depuración del código, los valores que se envían o reciben son incorrectos o las reglas de negocio no se aplican o se aplican de forma errónea.

Solución: Dedicar tiempo para conocer completamente nuestro entorno de desarrollo, así como las herramientas de depuración que proporciona, en el caso de los navegadores están disponibles diferentes plugins para depurar el código y solucionar los problemas de transferencia de datos y componentes cargados al ejecutar el proyecto.

3. Mostrar desinterés en la reunión diaria

Todos los días por la mañana se realiza una reunión diaria para conocer el avance del sprint, las historias por trabajar en el día y eliminar los obstáculos que impiden alcanzar los objetivos planeados. En ocasiones tenemos nuestra atención en otros pendientes y no prestamos atención a las intervenciones de los miembros del equipo, generando con esto una falta de visión del estado del proyecto, pérdida de conocimiento e ideas para conocer como eliminar obstáculos específicos y la oportunidad de integrarnos al equipo apoyando en la solución de estos.

Solución: Al iniciar el día de trabajo, debemos priorizar las actividades a realizar, mostrar una actitud participativa al formar parte de la reunión diaria, presentar la información del avance del día de anterior, lo planeado para el día actual y los obstáculos si hubiese que nos impiden avanzar. También dejar de lado todos los demás pendientes y concentrarse completamente en la reunión para absorber

las ideas de soluciones y apoyar con las nuestras a los demás miembros del equipo.

4. Perder de vista la fecha de entrega del sprint

Las actividades de un día de trabajo no siempre es escribir código, tenemos que asistir a diferentes tipos de reuniones con el equipo, clientes o directivos. Por otra parte, la procrastinación de las actividades a realizar para cumplir los objetivos del proyecto generan un efecto negativo en nuestro desempeño profesional, llegando hasta el extremo de una separación del proyecto por bajo rendimiento.

Solución: Reducir las distracciones al mínimo, concentrarse en las actividades a realizar y tener presente que las reuniones a pesar de que abarcaran una gran parte del día no son pretexto para cambiar la fecha de entrega, por lo que no debemos dejar tareas pendientes para el día siguiente, ya que cada día iniciamos con objetivos nuevos por alcanzar. En general el desarrollador debe tener conocimientos de la metodología ágil Scrum para cumplir con los sprints del proyecto.

5. No enviar reporte diario

Después de una jornada larga de trabajo, a menudo deseamos salir lo antes posible de la oficina para descansar y olvidar el estrés generado durante el día, concentrándonos en cerrar las actividades y olvidamos enviar el reporte de avance de las historias que desarrollamos. Esto genera llamadas de atención y roces con el líder del equipo, ya que afecta la visión general del estado del proyecto y retrasa la preparación de su informe a los clientes y directivos.

Solución: No importa que tan estresante fue el día de trabajo o sí el avance logrado no es el esperado, siempre tenemos que disciplinarnos en preparar el reporte lo más detallado posible para informar al líder del equipo el estado de

nuestras tareas y los obstáculos que impiden alcanzar los objetivos. Créelo el equipo, los directivos y el cliente lo agradecerán.

6. Comunicación deficiente con el equipo

Cuando recién iniciamos en el mundo del desarrollo de software con la participación en un proyecto como parte del equipo, nuestro mayor temor es que piensen que nos falta capacidad técnica. También la falta de experiencia en el trabajo en equipo imposibilita la comunicación con los demás miembros alejándonos de equipo en vez acercarnos para alcanzar los objetivos propuestos. Estas situaciones permiten que las tareas que no logramos realizar se atrasen mientras intentamos encontrar la solución de tareas que desconocemos por nosotros mismos, generando un efecto de bola de nieve que la mayoría de las veces no termina bien para nadie.

Solución: Debemos tener en cuenta que, al ser principiantes, en el equipo todos los miembros saben de nuestra condición y debemos mantener los canales de comunicación abierta para apoyarnos con los demás para encontrar soluciones rápidas a los problemas complejos que seguro nos tocarán todos los días hasta que la experiencia nos alcance y entonces las situaciones que nos quitaban el sueño serán simples anécdotas para recordar.

7. Falta de capacitación

El trabajo que realizamos en la industria del desarrollo de software, no se basa únicamente en un lenguaje de programación o tecnología específica, a lo largo de nuestra carrera profesional tendremos retos en los cuales la constante son tecnologías nuevas para nosotros y en estas asignaciones el error que cometemos con frecuencia es rechazar el reto argumentando el desconocimiento sobre el tema por el temor que genera en nosotros el fallar en el cumplimiento de los objetivos.

Solución: Debemos ser francos con respecto a nuestros conocimientos sobre el tema, recalcar que los nuevos retos son bienvenidos y pondremos todo nuestro tiempo en la investigación, desarrollo y cumplimiento de los objetivos trazados. También debemos aprovechar parte de nuestro tiempo libre en el aprendizaje de otras tecnologías, para obtener un conocimiento general de lo que sucede en la industria del TI en México y el mundo.

Capítulo 7 Conclusiones

El desarrollo de software es un área que tiene múltiples aplicaciones en todos los ámbitos de nuestra vida y está presente en todos los dispositivos que utilizamos en la actualidad. Las oportunidades para explotar profesionalmente este campo como Ingenieros en Redes son amplias y prometedoras, sin embargo, requiere capacitación adicional y gran esfuerzo para convertirse en desarrollador profesional.

Este trabajo presentó de manera general: procesos de desarrollo de software enfocados a la capacitación de nuevos talentos en JAVA y .Net, la administración de proyectos mediante la metodología ágil Scrum y errores de principiantes que normalmente cometemos al integrar equipos para realizar proyectos.

Los conocimientos adquiridos en este proceso de capacitación fueron importantes para mi integración con los equipos de desarrollo en plataformas tecnológicas distintas. El aprendizaje de HTML, CSS, JQuery y Javascript fue fundamental para realizar las actividades de diseño web que me asignaron en DNN, Sencha ExtJS, ASP.Net Web Forms, ASP .Net MVC with Razor, AngularJS.

La codificación del backend de todos los proyectos en los que participé se desarrollaron en C# utilizando OOP, éste fue el punto más complicado de mi experiencia en el desarrollo de software, debido al incipiente conocimiento que tenía en este lenguaje de programación, el cual contiene un amplio número de sentencias que facilitan la escritura de código. También el tema de la depuración del código representó un reto muy interesante al no estar familiarizado con las herramientas de desarrollo Web que facilitan la revisión de los errores en los navegadores como Internet Explorer, Mozilla Firefox y Google Chrome.

El manejo de BD con SQL Server y la configuración de servidores dedicados con Windows Server fueron temas que se facilitaron por los conocimientos recibidos en las

aulas en temas de Bases de datos y Sistemas Operativos de red. Sin embargo, no dejaron de presentar complicaciones al diseñar, normalizar y optimizar las BDs o configurar servidores desde cero en entornos administrados remotamente para hospedar BDs y sitios web.

El entorno de programación utilizado fue Visual Studio con .Net, las tecnologías empleadas para interacción con las BDs son Dapper ORM, Entity Framework Code First y Ninject DI.

El desarrollo de aplicaciones móviles en Xamarin fue otro de los retos interesantes en esta experiencia profesional, a pesar de que la codificación de los proyectos se realiza en .Net y C#, la compatibilidad de las sentencias del lenguaje es reducida, obligando a revisar las líneas de código que se escriben. El diseño de la UI de la aplicación que se realiza en XAML considera múltiples variantes de idioma, resolución y orientación de pantalla, colores, valores y diferentes pantallas. Las BD locales se manejan en SQL Lite y consumen datos de mediante un RESTful Web Service.

La experiencia profesional se enriqueció con la oportunidad de administrar proyectos con la metodología ágil Scrum que es un cambio de paradigma con respecto a la administración tradicional de proyectos de desarrollo de software, al integrar al cliente como parte del equipo desde el arranque del proyecto y programar entregables completamente funcionales para que el cliente evalúe y retroalimente con tiempos de respuesta cortos que agilizan la toma de decisiones y los ajustes de la dirección del proyecto, obteniendo con esto una completa satisfacción del cliente al concluir el proyecto en el tiempo y presupuesto estimados.

Bibliografía

1. **Frame, J. D.** *La nueva dirección de proyectos: herramientas para una era de cambios rápidos*. Barcelona, España : Granica, 2000.
2. **Gido, J. y Clements, J. P.** *Administración exitosa de proyectos*. México, D.F. : Cengage Learning Editores, S.A. de C.V., 2012.
3. **Ahmed, P. K., Sheperd, C. D., Ramos Garza, L., & Ramos Garza, C.** *Administración de la innovación*. Naucalpan de Juárez, Estado de México : Pearson Educación de México, S.A. de C.V., 2012.
4. **Cleland, D. I., & King, W. R.** *Manual para la Administración de Proyectos*. México : Compañía Editorial Continental, 2001.
5. **Chamoun, Y.** *Administración Profesional de Proyectos*. México, D.F. : McGraw-Hill/Interamericana Editores, S.A. de C.V., 2006.
6. **Leiva Mundaca, Ignacio y Villalobos Abarca, Marco.** Método ágil híbrido para desarrollar software en dispositivos móviles. *Ingeniare. Revista Chilena de Ingeniería*. [En línea] 2015. [Citado el: 27 de 11 de 2017.] <http://www.redalyc.org/articulo.oa?id=77241115016>. ISSN 0718-3291.
7. **Apiumhub.** Apiumhub.com. [En línea] 08 de 07 de 2018. [Citado el: 29 de 10 de 2018.] <https://apiumhub.com/es/tech-blog-barcelona/situacion-actual-del-software/>.

Glosario

Back End. Es la capa de acceso a datos de un software o cualquier dispositivo, que no es directamente accesible por los usuarios, además contiene la lógica de la aplicación que maneja dichos datos. El Backend también accede al servidor, que es una aplicación especializada que entiende la forma como el navegador solicita cosas.

Clase abstracta. Una clase que declara la existencia de métodos, pero no la implementación. Contiene métodos abstractos, métodos concretos y datos que serán implementados por las clases hijas.

Clase concreta. Clase que puede usarse para instanciar objetos y tiene implementaciones para todas sus operaciones.

Clase. Tipo definido que determina las estructuras de datos y operaciones asociadas con ese tipo. Las clases son como plantillas que describen como ciertos tipos de objetos están contruidos. Cada vez que se construye un objeto de una clase, estamos creando lo que se llama instancia de la clase. Por consiguiente, los objetos no son más que instancias de clases. En general, los términos objeto e instancia de una clase se pueden utilizar indistintamente.

Composición. Consiste en utilizar variables de instancia como referencia a los objetos. Pegar objetos de otras clases dentro de una nueva clase, mecanismo esencial para la reusabilidad del código.

Decorator. Patrón de diseño para solucionar problemas donde la jerarquía con subclasificación no puede ser aplicada, o se requiere de un gran impacto en todas las clases de la jerarquía con el fin de poder lograr el comportamiento esperado. Decorator permite al usuario añadir nuevas funcionalidades a un objeto existente sin alterar su estructura, mediante la adición de nuevas clases que envuelven a la anterior dándole funcionamiento extra.

Diferencia entre clase abstracta e interfaz. En la clase abstracta además de los métodos abstractos también es posible definir datos y métodos concretos que serán heredados por las clases hijas, por el contrario, la interfaz solo define comportamiento; es decir, aísla conceptualmente el comportamiento y no contamina con datos o métodos innecesarios a las clases que la implementan.

Encapsulamiento. Permite a los diseñadores de tipos de datos determinar que miembros de los tipos pueden ser utilizados por otros programadores y cuáles no. Se pueden ocultar todos los detalles relativos a su implementación interna y dejar visibles aquellos que puedan usarse con seguridad. La encapsulación se consigue agregando modificadores de acceso a las definiciones de los miembros y tipos de datos.

Factory Method. Patrón de diseño que permite la creación de objetos de un subtipo determinado a través de una clase Factory. Es especialmente útil cuando no sabemos, en tiempo de diseño, el subtipo que vamos a utilizar o cuando queremos delegar la lógica de creación de los objetos a una clase Factory. Utilizando este patrón creamos instancias dinámicamente mediante la configuración estableciendo cual será la implementación a utilizar en un archivo de texto, XML, properties o cualquier otra estrategia.

Front End. Es la parte de un programa o dispositivo a la que un usuario puede acceder directamente. Son todas las tecnologías de diseño y desarrollo web que corren en el navegador y que se encargan de la interactividad con los usuarios.

Herencia. Es cuando una clase hereda sus propiedades y métodos a otra clase, ésta segunda se convierte en una subclase y la primera en clase superior o clase padre.

Interfaz. Medio para definir comportamientos, los cuales se agrupan dentro de la interfaz y están relacionados entre sí lógicamente o funcionalmente. Los comportamientos definidos en la interfaz son métodos abstractos únicamente.

Métodos. Describen el comportamiento asociado a un objeto. Representan las acciones que se pueden realizar por un objeto o sobre un objeto.

Objeto. Los objetos son tipos abstractos de datos, encapsulan datos y funciones que operan sobre esos datos.

Patrón de diseño. Es la solución a un problema de diseño, el cual debe haber comprobado su efectividad resolviendo problemas similares en el pasado, también tiene que ser reutilizable, por lo que se deben poder usar para resolver problemas parecidos en contextos diferentes.

Polimorfismo. Una clase puede utilizarse como más de un tipo; puede utilizarse como su propio tipo, o como el tipo de su clase base.

Propiedad. También llamados atributos, describen la abstracción de características individuales que poseen todos los objetos. Describen el estado del objeto y consta de dos partes: un nombre y un valor.

Singleton. Este patrón de diseño recibe su nombre debido a que sólo se puede tener una única instancia para toda la aplicación de una determinada clase, esto se logra restringiendo la libre creación de instancias de esta clase mediante el operador new e imponiendo un constructor privado y un método estático para obtener la instancia. La intención de este patrón es garantizar que solamente pueda existir una única instancia de una determinada clase y que exista una referencia global en toda la aplicación.

Strategy. Patrón de diseño permite establecer en tiempo de ejecución el rol de comportamiento de una clase, se basa en el polimorfismo para implementar una serie de comportamientos que podrán ser intercambiados durante la ejecución del programa, logrando con esto que un objeto se pueda comportar de forma distinta según la estrategia establecida.

Abreviaturas

AngularJS. Angular Java Script.

ASP.Net. Es un modelo de desarrollo Web unificado que incluye los servicios necesarios para crear aplicaciones Web empresariales con el código mínimo.

Bootstrap. Framework desarrollado y liberado por Twitter que tiene como objetivo facilitar el diseño web.

C#. Lenguaje de programación orientado a objetos desarrollado y estandarizado por Microsoft como parte de su plataforma .NET.

CSS. Cascading Style Sheets (Hojas de Estilo en Cascada)

DB. DataBase (Base de datos).

Entity Framework. Conjunto de tecnologías de ADO.NET que permiten el desarrollo de aplicaciones de software orientadas a datos

HTML. HyperText Markup Language (lenguaje de marcas de hipertexto)

Java. Lenguaje de programación

Javascript. Lenguaje interpretado orientado a las páginas web.

Jqgrid Plugin.q. Plugin de jquery que permite crear tablas dinámicas

JQuery. Framework de JavaScript

MVC. Model View Controller (Modelo Vista Controlador).

MyBatis. Herramienta de persistencia Java que se encarga de mapear sentencias SQL y procedimientos almacenados con objetos a partir de ficheros XML o anotaciones.

MySQL. Sistema de administración de bases de datos (Database Management System, DBMS) para bases de datos relacionales.

Ninject DI. Ninject Dependency Injection (Inyector de Dependencias Ninject).

OOP. Oriented Object Programming (Programación Orientada a Objetos)

Oracle 10g+. Base de datos Oracle 10g.

PM. Project Manager (Administrador de Proyectos)

Razor. Es una sintaxis de programación ASP.NET se utiliza para crear paginas dinámicas con C# o con Visual Basic.NET

Scrum. Framework de desarrollo ágil más extendido. Se trata de un método iterativo e incremental en el que se divide el desarrollo de un producto en ciclos o sprints.

Service REST. Service Representational State Transfer (Servicios de Transferencia de Estado Representacional).

Spring MVC. Framework alternativo al stack de tecnologías estándar en aplicaciones JavaEE.

Spring Security. Framework que permite gestionar todo lo relativo a la seguridad de una aplicación web.

SQL. Structured Query Language (Lenguaje de Consulta Estructurada)

UWP. Universal Windows Platform (Plataforma Universal de Windows).

Web API. Web Application Programming Interface (Interfaz de programación de aplicaciones Web).

Xamarin. Herramienta para desarrolladores de aplicaciones móviles.

Xamarin.Forms. Herramienta para aplicaciones móviles que pueden compartir gran parte del código de su interfaz de usuario y conservar la apariencia nativa de la plataforma de destino

XAML. EXtensible Application Markup Language (Lenguaje Extensible de Formato para Aplicaciones).