

UNIVERSIDAD ESTATAL A DISTANCIA
VICERRECTORIA ACADEMICA
ESCUELA DE CIENCIAS EXACTAS Y NATURALES
CARRERA INGENIERÍA INFORMATICA

PROYECTO No 1

MODALIDAD ESCOGIDA: PROYECTO

PROYECTO #1 PARA EL CURSO

DE Ingeniería del software

PABLO ANDRÉ VALENCIANO BLANCO

1-1572-0043

CENTRO UNIVERSITARIO DE HERERIA

PAC: 2024-1

CIUDAD: HEREDIA

Contenido

Introducción.....	3
Parte 1. Cuadro Comparativo, metodologías tradicionales y ágiles.....	4
Parte 2. Metodología Proceso Unificado de Desarrollo de Software (RUP).....	5
Parte a. Partes que la componen.....	5
Parte b. Las disciplinas que contempla Proceso Unificado de Desarrollo de Software.....	6
Parte 3. Técnica Entrevista Cerrada.....	8
Parte 4. Tarifa diferenciada, un buen plan de trabajo.....	9
Parte a. Requerimientos funcionales (Separación por tipo de combustión).....	9
Parte b. Impacto de la implementación propuesta: Sistema de estacionamiento de Torre.....	9
Parte 5. Programación Extrema (XP).	10
Parte a. Concepto programación Extrema.....	10
Parte b. Valor que compone el marco de trabajo de programación extrema.....	10
Parte c. Cuadro comparativo. Ventajas y desventajas de aplicar el marco trabajo programación extrema.....	11
Conclusión.....	12
Referencias.....	13

Tabla de Figuras

Figura 1.....	5
---------------	---

Tabla de Tablas

Tabla 1.....	4
Tabla 2.....	11

Introducción

El siguiente documento dará por presentado el primer proyecto del curso de Ingeniería de software, el cual va a ser realizado por el estudiante y se apoyaran las explicaciones brindadas con fuentes bibliográficas las cuales se van a documentar con APA versión 7.

En primera instancia se realizó la lectura del caso de estudio: Sistema de Estacionamiento en Torres, la cual se comentará de ella durante todo el cuatrimestre a lo largo de los proyectos que se van a ir trabajando.

Este primer proyecto consta de contestar las preguntas, las cuales serán divididas en partes y cada uno de estas, según su complejidad, podría llegar a subdividirse a más subpartes, para poder lograr el objetivo de que el estudiante posea los conceptos y se apoye en material auxiliar para fundamentar las ideas que responderás las cuestiones. Se intentará para que la documentación no sea tan pasada para el lector, se escribirá en prosa en búsqueda de una fácil comprensión.

El proyecto, posee los siguientes objetivos específicos:

- Que el estudiante amplíe sus conocimientos en las distintas metodologías existentes.
- Que el estudiante amplíe sus conocimientos acerca del Marco de Trabajo de Programación Extrema (XP).
- Que el estudiante amplíe sus conocimientos acerca del proceso unificado (RUP) a través de la investigación de las fases y disciplinas que la componen.
- Que el estudiante amplíe sus conocimientos acerca del manejo de las distintas técnicas para el levantamiento de requerimientos.

Una vez adquirida esta información será fundamental la opinión y la apropiación de dichos conceptos, para poder demostrar por medio de conclusiones que se logra el objetivo general; el cual consiste en que a través de un caso de estudio se entienda la importancia y metodologías de realizar una buena gestión de la ingeniería y desarrollo de software que generen una solución real a problemas típicos, y convertir ideas a elementos tangibles y con alto grado de utilidad.

Parte 1. Cuadro Comparativo, metodologías tradicionales y ágiles.

	Metodologías Tradicionales	Metodologías Ágiles
Ventajas		
1.	Al ser en cascada, los procesos tiene un principio y fin, que habilita el siguiente proceso, hasta dar por finalizado el proceso, siendo bastante claro y ordenado	El objetivo final es flexible, el cual evoluciona a medida que va avanzando el proyecto.
2.	La precisión al momento de calendarizar fechas para cada objetivo son más precisas que las metodología ágiles	Hay una participación activa de parte del cliente o usuario final, el cual va agregando las variables para cumplir el objetivo
3.	Se prioriza la documentación de cada proceso, con el objetivo de registrar y comentar la forma en cómo se llegó y las metas que se lograron cumplir durante el proyecto	El compromiso entre los participantes del desarrollo, es de mayor calidad y hacen el proyecto de ellos antes que de un gerente.
Desventajas		
1.	Las revisiones de pruebas de calidad son de las últimas fases y de encontrarse un bug, puede conllevar retrasos que depende del impacto	Tiene una planificación poco concreta, debido a la constante reestructuración de las nuevas variables
2.	El cliente no se integra al proceso de producción dando retroalimentación hasta que el producto es entregado	Se desvaloriza la documentación, dando más prioridad a los entregables del proyecto
3.	Es bastante rígido y muy complicado de cambiar, si los requerimientos cambian. Que conllevan hasta reiniciar todo el procedimiento	Los equipos son pequeños, y usualmente sus integrantes van cambiando según las necesidades, lo cual implica que cada uno de los miembros debe estar capacitado de realizar distintas tareas

Tabla 1. Metodologías Tradicionales vs Ágiles. Información extraída de LinkedIn(2022)

Parte 2. Metodología Proceso Unificado de Desarrollo de Software (RUP)

Las fases que la componen son 4, las cuales se muestran en la siguiente imagen y la acompañan las disciplinas en las cual se elabora cada proyecto, y el esfuerzo en el tiempo en cada una de ellas. Toda la siguiente información es descrita luego de la lectura extraída de ECURED (2010)

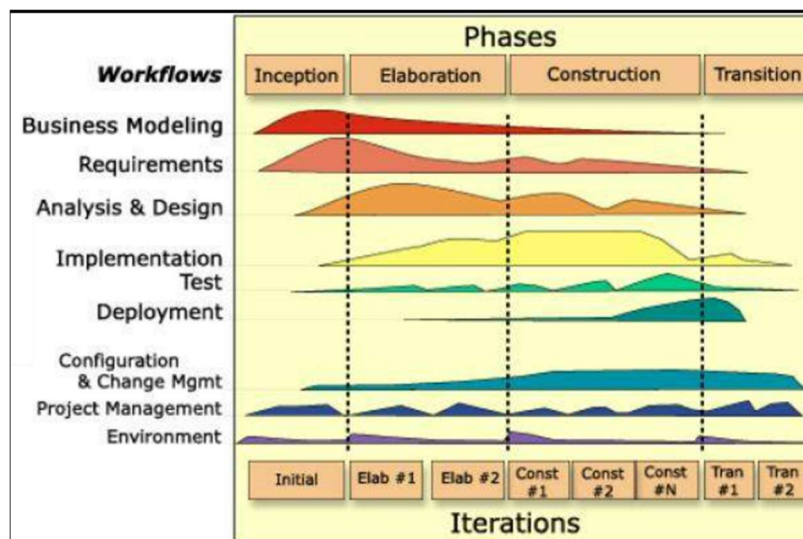


Figura 1. Esfuerzo de cada disciplina en cada fase del proyecto

Parte a. Fases que componen la metodología del Proceso Unificado de Desarrollo de Software

La primera de las fases es la fase de concepción, la cual establece los cimientos para el éxito del proyecto; tales como definir la visión, los objetivos y el alcance que este puede llegar a tener. Es de suma importancia que durante esta fase se definan requerimientos y los presupuestos para el cumplimiento del objetivo. Es la única fase que no requiere de un entregable y se da los primeros acercamientos de posibles soluciones

La siguiente fase es la de elaboración, en la cual se terminan de definir los casos de uso del proyecto y se definen las herramientas de desarrollo que mejor se apropien a los apropiados de desarrollar un primer entregable, el cual entra en una etapa de reiteraciones y revisiones de nuevos posibles casos y/o requerimientos sujetos a resolver de mejor manera los objetivos. Se da entregable de fase temprana.

La posterior fase es la de construcción en la cual todos los implicados realizan cada uno su parte, para poder cumplir en cada aspecto la mejor forma de visionar el éxito del proyecto. Esta fase está compuesta de múltiples iteraciones, donde se van cumpliendo los casos de uso y acorde a la prioridad de cada uno y su funcionalidad para el proyecto.

La última fase, es la fase de transición, la cual inicia con la primera versión que cumple todos los casos de uso, hasta la fecha propuestos, a esta versión se le denomina versión beta y se va realizando mejoras hasta llegar a una fase de producción, en la cual se prepara para una revisión final y solucione los fallos faltantes.

Parte b. Las disciplinas que contempla Proceso Unificado de Desarrollo de Software

- **Modelo de Negocios:** Es la disciplina que se encarga de comprender la estructura y operaciones del negocio relacionado con el sistema a desarrollar. El objetivo es establecer una base sólida para la identificación de requisitos y el software que se alinee al proyecto.
- **Requisitos:** Durante el proceso se debe identificar, analizar, documentar y validar los requisitos de los sistemas. Se deben estudiar los casos de uso para describir las interacciones entre los sistemas y los usuarios. Definiendo así, los requisitos funcionales y no funcionales.
- **Diseño:** Es donde se lleva la transformación de los requisitos a un sistema estructurado. Se definen todos los bloques desarrollo desde el mas alto nivel, hasta los componentes que lo incorporan. Se asignan las responsabilidades a cada equipo.
- **Implementación:** Se traduce el diseño en código ejecutable. Se realiza la programación, las pruebas singulares para cada caso y la integración de los componentes. Con el objetivo de presentar un código eficiente y cumpla los requisitos actuales.
- **Pruebas:** Se encarga de validar que cumpla los requisitos establecidos. Se realizan pruebas para todos los componentes de forma exhaustiva y que el código no sea interrumpa por “corner cases”, además se debe validar la integración y que todos los módulos se comporten de la manera deseada.

- **Despliegue:** Una vez satisfecha las prueba, se integra al entorno de producción. Esta fase debe ser previamente planificada y que los principales integrantes estén preparados en caso de una mala funcionalidad, dando un buen manejo de posibles problemas que pueden surgir durante la implementación.
- **Gestión de Proyectos:** Se centra en la planificación, seguimiento y control del proyecto. Es donde se definen los objetivos, la correcta asignación de recursos, la gestión de riesgos y la supervisión del proyecto. Este con la responsabilidad de cumplir los plazos y la calidad prometida.
- **Gestión del Entorno:** Son la parte encargada de validar, responder y actuar que los recursos sean accesibles. La cual debe brindar mantenimiento todo aquel entorno que deba ser revisado y de ser necesitado remplazar, para cumplir las metas del proyecto.
- **Gestión de Configuración y Cambios:** Es la encarga de revisar que todo se encuentre debidamente documentado y se revise los cambios que suceden durante el transcurso del proyecto. Debe controlar las versiones y la gestión de cambios para garantizar que el software se desarrolle de manera controlada.

Parte 3. Técnica Entrevista Cerrada

De forma simple, “el uso de las preguntas cerradas está más extendido cuando se desea obtener información cuantificable y se quiere asignar un valor a cada respuesta o para restringir respuestas.” Este concepto dado por la pagina Sesame (2022).

Las 10 preguntas escogidas para realizar la metodología las entrevistas son las siguientes:

1. ¿Usted posee un auto, el cual debe ser parqueado en la UNED?
2. ¿Cuántas veces a la semana debes usar el parqueo de la UNED?
3. ¿Consideras complicado al momento de llegar al parqueo de la UNED conseguir un espacio disponible?
4. ¿Se te facilita el uso de aplicaciones en el teléfono con el fin de facilitar procesos cotidianos?
5. ¿Te parece correcto que, por medio de una app, se pueda reservar tu campo en los parqueos, para que cuando llegues conozcas el espacio, el cual debes ubicarte?
6. ¿Cuál sería para ti un tiempo prudente del tiempo de reserva?
7. ¿Crees oportuno tener la chance de pagar en efectivo, electrónico o ambas?
8. ¿Debe existir un pago diferenciado para estudiantes y profesores identificados?
9. ¿La hora pico, debería estar diferenciada, dado que aumentaría la demanda?
10. ¿Son las reservas una solución real para gestionar de forma adecuada los espacios de parqueo en la UNED?

La importancia de la entrevista del tipo encuesta cerrada, la cual realiza preguntas simples y funcionales para lograr ver el alcance y demostrar la importancia que el proyecto va a tener y si el impacto es por lo menos lo que equivale el gasto. Con esta técnica se obtienen datos fiables y válidos, aunque a cambio de no permitir que el encuestado se exprese libremente o se calle, en relación con los temas que se consideran relevantes en esa situación.

Parte 4. Tarifa diferenciada, un buen plan de trabajo

Parte a. Requerimientos funcionales (Separación por tipo de combustión)

El primer requisito funcional, es que se debe diseñar un apartado, donde la persona debe ingresar sea la matrícula, por medio del registro de placas saber qué tipo de carro es que la posee, validar si es eléctrico, híbrido, gas metano o de combustión.

El siguiente requisito funcional, es una segunda opción a la primera, la cual puede ser ingresar todos los días, el vehículo que va a ingresar, para que así el consumidor pueda aprovechar de forma satisfactoria el precio diferenciado.

El tercer requisito, el cual puede o no ser implementado con software, sea con cámaras o lector de matrícula, el cual debe validar que la persona que reservo el sitio asignado, realmente cumpla las características de usar energía sostenible

El cuarto requisito, es que si se decide registrar un vehículo en la aplicación. Este debe tener la posibilidad de cambiar sus características, ya que automóviles previamente de solo gasolina, se pueden transformar a un híbrido, e igual manera revertir el cambio.

El último de los requisitos, es ingresar es aplicar estos nuevos precios y ajustarlos al resto de requisitos (hora pico, mensualidad), sea con multiplicadores, porcentajes de descuento o mantener precios para incentivar el uso de energía renovables.

Parte b. Impacto de la implementación propuesta: Sistema de estacionamiento de Torre

De manera general, una implementación como esta siempre es positiva, por dos grandes razones: novedoso y permite al consumidor más opciones siendo la herramienta con más casos de posible uso. La novedad siempre va atraer y abre las posibilidades a que otras implementaciones pueden basarse en los programas ya contruidos. El otro factor positivo, es la amplitud del negocio la cual va a incentivar a que pasarse a energías renovables va a ser mucho más rentable, sea comprando un vehículo nuevo o transformando el actual, a modo símil, es como cuando un restaure ofrece opciones vegetarianas y de gluten. El impacto negativo, es mas relacionado con el proyecto y al ampliar los casos de uso, se debe de alguna forma o otra, cumplir los nuevos requerimientos y se debe saber que estas nuevas integraciones implican tiempo y costo.

Parte 5. Programación Extrema (XP).

Parte a. Concepto programación Extrema

Según el concepto extraído y resumido por la revista digital IONOS (2019), el concepto de programación extrema se refiere a “la aplicación más radical del desarrollo ágil de software. Dicho de otra manera: es probable que no haya ningún método, y mucho menos una forma tradicional de programación, más ágil que el XP. En este contexto, la programación extrema se diferencia concretamente del desarrollo en cascada.” Este se compone de valores, principios y técnicas, según lo que se desee la cantidad puede variar de cada uno.

Parte b. Valor que compone el marco de trabajo de programación extrema

Usando la misma guía de IONOS (2019), poseemos 5 valores: Comunicación, Sencillez, Retroalimentación, Valentía y Respeto.

La comunicación engloba tanto la capacidad de los miembros del equipo realizador, sino también incorpora a los clientes. De esta forma si todos los implicados están en contacto de forma permanente se pueden evitar y detectar rápidamente los malos entendidos.

Mientras tanto la sencillez, busca darles enfoque a los factores necesarios, el plan es no distraerse con cuestiones secundarias. Se concentra solo en desarrollar solo las funciones necesarias en cada momento. Además, un producto es más fácil de manipular, en caso de ser retomado.

Otro valor importante es la retroalimentación, la cual el cliente posee un canal directo de expresar críticas. Estas posibles mejoras deben ser en pasos pequeños, que no sean un cambio radical del proyecto, el cual lleve a fortalecer los requerimientos, no a confundir al desarrollo.

El siguiente valor es la valentía, la cual se atribuye en reconocer los fallos, encontrar problemas y reportarlos, buscar modos de cómo solucionar. Ser capaz de como equipo que ser capaz de cuestionar métodos de trabajo, aceptar críticas y si es necesario escribir código desde cero se debe hacer.

El último de los valores, en el cual todos los canales deben poseer ese modo de operar es por medio del respeto, donde todos los involucrados, están en su posición en busca de darle solución a cada porción del proyecto. Donde todos los comentarios se deben hacer en forma de ayudar, y si se están solicitando consultas o recursos, es con el fin de darle pronta solución y avanzar de forma segura en el proyecto.

Parte c. Cuadro comparativo. Ventajas y desventajas de aplicar el marco trabajo programación extrema.

Ventajas	Desventajas
1.Ausencia de trabajos de programación innecesarios. Enfocada en funcionalidades primarias.	1.Los desarrolladores de alta capacidad y cumplan las métricas son usualmente caros.
2.Ausencia de horas extra, cada miembro gestiona su propio tiempo.	2.Requiere control de versiones, y cada equipo debe encargarse de documentar esos cambios.
3. Código de comprensión sencilla en todo momento.	3.El cliente se implica en el esfuerzo, y en algunos casos llega a ser invasivo durante el desarrollo.

Tabla 2. Ventajas y Desventajas de la Programación Extrema (XP)

Conclusión

Ya llegados a este punto, se darán las respectivas conclusiones del trabajo realizado, la primera conclusión es sobre la primera parte donde se compara las metodologías tradicionales y las metodologías ágiles, en ella nos damos que las metodologías tradicionales, están enfocadas en la planificación y documentación, mientras las metodologías ágiles prefieren darle prioridad al desarrollo y realización de pruebas, ambos métodos válidos y que, según el tipo de proyecto, uno u otro es mejor.

En la segunda parte, se discutió sobre el RUP, el cual de forma colectiva y unificada los miembros de cada disciplina realizan enfoques y labores durante cada fase desde que se da un inicio donde se preparan los requerimientos, a través de un desarrollo se entregan las primeras versiones, pasa por revisión y por ultimo llega a producción, dando un inicio y un fin, donde todos en algún momento aportaron a la entrega final.

La tercera parte, es una metodología llevada al caso de estudio, que se va a seguir trabajando en los siguientes proyectos, y permite consultar a los interesados o beneficiarios sobre los objetivos y alcance que el proyecto puede proveer, por medio de una encuesta que realiza preguntas cerradas y da un interés mayor de como dicha aplicación debería ser diseñada.

Una vez realizada la encuesta, nos damos cuenta que se podría trabajar en más posibles forma de incentivar a los beneficiarios de usar energías limpias, por medio de reducciones al pago del uso de la red de datos para el parqueo cuando utilizan un vehículo de ciertas características de combustión, y se analizan los respectivos pros y contras de realizar dicha mejora. Los cuales, analizándolo es una mejora adecuada y una inversión que ayuda no solo al ambiente, sino a la ideología de que las nuevas tecnologías, deben ir enfocadas a proteger el ecosistema.

Por último, vimos la programación extrema con ello nos da a entender que cuando se consigue el personal adecuado, hay una buena comunicación con el cliente y los trabajos se desarrollan de forma sencilla, se logran proyectos con márgenes de calidad y producción mejores que el resto de metodologías, lo más complicado para llevar este

tipo de esfuerzo lo que hace falta es entendernos sin faltar en ningún momento el respeto y atendiendo las dificultades cuando estas se den.

Referencias

LinkedIn (2022). Diferencias entre metodologías ágiles y tradicionales: ventajas y desventajas. Extraído de: <https://www.linkedin.com/pulse/diferencias-entre-metodolog%C3%ADas-%C3%A1giles-y-tradicionales-ventajas-/?originalSubdomain=es>

Ecured (2010). Proceso unificado de desarrollo. Extraído de: https://www.ecured.cu/Proceso_unificado_de_desarrollo

Sesame (2022). Preguntas cerradas. Extraído de: <https://www.sesametime.com/assets/diccionario/preguntas-cerradas/#:~:text=Las%20preguntas%20cerradas%20en%20un,responder%20de%20manera%20m%C3%A1s%20amplia.>

Raeburn, A. (2022). La programación extrema (XP) produce resultados, pero ¿es la metodología adecuada para ti? Extraída de: <https://asana.com/es/resources/extreme-programming-xp>

IONOS (2019). Extreme Programming: desarrollo ágil llevado al extremo. Extraído de: <https://www.ionos.es/digitalguide/paginas-web/desarrollo-web/extreme-programming/>