

Universidad Estatal a Distancia

Vicerrectoría Académica

Escuela De Ciencias Exactas y Naturales

Carrera de Diplomado en Ingeniería Informática

Asignatura: Ingeniería del Software

Código: 03300

Proyecto 1

Estudiante:

Francisco Campos Sandi

114750560

Sede: San Vito

Grupo 01

Tutora: Iliana Segura Picado

III Cuatrimestre 2024

Contenido

Introducción	3
Desarrollo	4
Tabla 1 Cuadro comparativo de metodologías	8
Tabla 2 Cuadro Metodología Ágil Scrum	12
Conclusión	13
Referencias	15

Introducción

El presente trabajo aborda el desarrollo e implementación de un sistema para CleanCarPlus, una solución innovadora en la administración de servicios de limpieza de vehículos. Este sistema tiene como objetivo mejorar la gestión de los servicios ofrecidos por el lavacar, facilitando tanto el registro de clientes y vehículos como la programación de citas.

La importancia de este trabajo radica en la creciente demanda de servicios de limpieza de vehículos, donde la eficiencia y la atención al cliente son factores clave para el éxito de un negocio. La implementación de un sistema automatizado no solo permitirá un mejor seguimiento de los servicios prestados, sino que también proporcionará a los usuarios internos las herramientas necesarias para llevar a cabo sus tareas de manera más organizada y efectiva.

En el desarrollo del trabajo se incluyeron varias fases, comenzando por el análisis de requerimientos y el diseño de un sistema que contemple los módulos necesarios para una operación eficiente. Se detallaron las funcionalidades específicas que cada módulo debe ofrecer, como el registro de ingresos de vehículos, la gestión de servicios y el seguimiento del estado de los mismos. Además, se contemplaron aspectos no funcionales, como la seguridad de los datos y la escalabilidad del sistema, asegurando que este pueda adaptarse a futuras necesidades.

El enfoque adoptado en este trabajo se centra en la metodología ágil, que permite una entrega incremental de los resultados y facilita la adaptación a cambios durante el proceso de desarrollo. Esta metodología es adecuada para el contexto del proyecto, dado que permite ajustar el sistema a las necesidades del cliente en tiempo real. A través de esta aproximación, se busca no solo cumplir con los requisitos iniciales, sino también garantizar que el producto final se ajuste a las expectativas y demandas del mercado.

Desarrollo

1. Tomando en cuenta el avance en la tecnología qué tipo de Desarrollo de Software está más en tendencia y por qué. Sea amplio en la respuesta con al menos 10 líneas y justifíquela con algún ejemplo, en donde el ejemplo puede ser algún software o empresa a nivel mundial.

En la actualidad, el desarrollo de software en tendencia es el de desarrollo de código abierto, dado que dicho modelo ha ganado popularidad por el método que involucra a otros desarrolladores en comunidades a nivel mundial y que puedan colaborar en proyectos que han cambiado la humanidad como se menciona en la siguiente página consultada su definición, “los proyectos de código abierto son aquellos que son desarrollados por un gran grupo de personas que contribuyen a la comunidad y son de dominio público. Dentro de los proyectos de software de código abierto que permiten al usuario crear sus propias aplicaciones” (*¿Qué son los proyectos de código abierto?*, 2021). Uno de los ejemplos más radicales en el cambio de la informática es el proyecto de Linux que se usan en muchas plataformas como en equipos de computación, la mayoría de servidores en el mundo, aplicaciones en sistemas de teléfonos celulares basados en Android que usan Linux.

El desarrollo de proyectos de código abierto genera muchos cambios en el uso colaborativo en el cual lo realizan personas del todo el mundo con la finalidad de poder resolver problemas que hay en nuestro entorno, así como el proyecto de Linux hay muchos más que gracias al ser “Open Source” nos vemos beneficiados en muchos ámbitos

2. De acuerdo con la documentación consultada cite y explique las Metodologías de Desarrollo de Software existentes y de una breve explicación en al menos 5 líneas de texto para cada una de ellas

Las dos metodologías de desarrollo de software de acuerdo al libro de texto del curso son:

1. Metodologías conductivas por planes:

“Se trata de procesos lineales que marcan los ciclos de programación de manera estricta. Antes de iniciar cada etapa se debe haber terminado la anterior. En este sentido, es una metodología poco flexible a la hora de realizar ajustes y en el desarrollo del proyecto”.(“Power up Your Workflow”, 2023)

Lo cual son metodologías basadas en un enfoque secuencial y estructural en cada fase de su desarrollo, cada etapa se va concretando detrás de cada una completada, no se puede saltar una etapa sin tener terminan como lo menciona (Pantaleo & Rinaudo, 2015, p. 23)

Planificación predictiva

Resistencia a los cambios

Orientadas a los procesos

Negociación de contratos

Las metodologías conducidas por planes tienen enfoques para el desarrollo de software los cuales son procesos que siguen una estructura dada en una secuencia de procesos para ir completando cada etapa lo cual lo hace menos flexible para no realizar un etapa antes de empezar otra, dicha metodología es ,uy útil cuando ya se tienen los requerimientos están viene definidos, de lo contrario puede volverse más costoso o complejo si no se está definido.

2. Metodologías Ágiles

“En la actualidad, las metodologías ágiles de desarrollo de software son ampliamente utilizadas debido a su alta flexibilidad y capacidad para adaptarse rápidamente a los cambios” (Valtx, 2023)

Las cuales por su nombre se tratan que son más flexibles al ir completando las etapas, las cuales se pueden desarrollar sin tener un orden específico, lo cual las hace más efectiva en los equipos de desarrollo las cuales no requieren un orden específico como la metodología anterior, lo cual el trabajo se reparte entre los equipos y no depende de que se termine la etapa anterior para poder avanzar, lo cual la hace un metodología en la que se puede desarrollar proyectos con un poco más de rápidos.

Estas metodologías varían en cuanto a flexibilidad, control y tiempo de entrega, por lo que la selección de la más adecuada dependerá de la naturaleza y las exigencias específicas del proyecto.

3. Mediante un cuadro comparativo, determine los tipos de metodologías conducidas por los planes. Dicho cuadro comparativo debe incluir una columna con una breve descripción en al menos 3 líneas de texto, otra columna con una ventaja y una última columna con una desventaja para cada una.

Metodología	Descripción	Ventaja	Desventaja
Cascada	Es un enfoque secuencial donde el desarrollo sigue una serie de fases: análisis de requisitos, diseño, implementación, pruebas, despliegue y mantenimiento. Cada fase debe completarse antes de pasar a la siguiente.	Facilita la gestión de proyectos grandes donde los requisitos están bien definidos y no suelen cambiar.	Es rígida y difícil de adaptar si surgen cambios en los requisitos durante el desarrollo, lo que puede generar retrasos.
Desarrollo Rápido de Aplicaciones (DRA)	Es un desarrollo rápido y ágil en la que se pueden llevar diferentes procesos en conjunto del desarrollo del proyecto y con los diferentes equipos de trabajos, así se reparten las responsabilidades en cada equipo y no en una sola área y que los proyectos sean entregados en periodos cortos	Con buenos requerimientos se logran entregas en periodos muy cortos	Al trabajar con varios equipos simultáneamente se requieren muchos recursos
Modelo en Espiral	Combina aspectos del desarrollo iterativo con el enfoque sistemático de las metodologías tradicionales. Se centra en la evaluación de riesgos y permite ajustes en cada ciclo o iteración.	Permite realizar mejoras incrementales y evaluar riesgos antes de cada iteración, lo que ayuda a identificar problemas potenciales de forma temprana.	Es más complejo y costoso en términos de tiempo y recursos debido a la constante evaluación y gestión de riesgos en cada fase del proyecto.
Modelo Incremental	Se divide el desarrollo del proyecto en pequeñas versiones o incrementos, donde cada incremento es funcional y aporta valor al producto final, completando características adicionales en cada versión.	Facilita la entrega continua de funcionalidad útil a los usuarios, lo que permite una retroalimentación más temprana.	La integración de los incrementos puede ser complicada si no se planifica adecuadamente, generando inconsistencias en las versiones anteriores.
Modelo de Prototipado	Se crean prototipos del sistema para que los usuarios puedan interactuar con ellos y proporcionar retroalimentación antes de	Mejora la comunicación con el cliente al permitirle visualizar el producto final antes	Puede generar malentendidos si el cliente espera que el prototipo tenga el mismo nivel de

	que se desarrolle el producto final.	de su implementación completa.	funcionalidad o calidad que el producto final.
--	--------------------------------------	--------------------------------	--

Tabla 1 Cuadro comparativo de metodologías

4. Para un plan de proyecto, describa las fases de un flujo de trabajo para una metodología conducida por los planes. De acuerdo con su criterio cuál de las fases indicadas es clave para que el proyecto cumpla a cabalidad con las necesidades del cliente Lavacar CleanCarPlus. Justifique su respuesta en al menos 10 líneas de texto. En una metodología conducida por los planes, el flujo de trabajo típico sigue una serie de fases secuenciales bien estructuradas. Estas fases son:
- **Recopilación de Requisitos:** En esta fase, se identifican y documentan las necesidades del cliente y las funcionalidades deseadas del sistema. Es crucial establecer una comunicación clara con el cliente para entender sus expectativas.
 - **Análisis:** Aquí se realiza un análisis detallado de los requisitos recopilados. Se identifican las soluciones técnicas y se evalúa la viabilidad de cada una.
 - **Diseño:** En esta fase se elabora un diseño detallado del sistema, tanto a nivel de arquitectura como de interfaces de usuario, asegurando que cumpla con los requisitos definidos.
 - **Implementación:** Consiste en el desarrollo del código del sistema según el diseño previamente aprobado.
 - **Pruebas:** Se realizan pruebas exhaustivas para asegurar que el sistema cumple con los requisitos y no presenta errores. Se prueban tanto las funcionalidades individuales como el sistema completo.
 - **Despliegue:** Una vez aprobadas las pruebas, se instala el sistema en el entorno de producción.
 - **Mantenimiento:** Después de la implementación, se realizan ajustes o correcciones necesarias basadas en el uso real del sistema.

La fase clave para que el proyecto CleanCarPlus cumpla con las necesidades del cliente es la Recopilación de Requisitos. En este proyecto, es fundamental captar con precisión todas las necesidades del cliente y definir claramente los servicios que el sistema debe ofrecer, como la gestión del ingreso de vehículos, la integración de pagos electrónicos, la funcionalidad de notificaciones y el seguimiento del estado de los servicios.

Si esta fase no se realiza adecuadamente, es probable que el sistema no cumpla las expectativas, lo que podría llevar a retrasos y modificaciones posteriores. Entender con claridad cómo el lavacar quiere gestionar sus operaciones es esencial para que las siguientes fases se alineen con los objetivos del proyecto y para asegurar que la solución desarrollada sea funcional y satisfactoria para el cliente.

5. Cite las fases del Proceso Unificado de Desarrollo y explique por cada fase cuál es su objetivo en el ciclo de vida de un proyecto

El Proceso Unificado de Desarrollo (UP) es un marco iterativo e incremental que divide el ciclo de vida del desarrollo de software en cuatro fases: concepción, elaboración, construcción y transición. Cada una de estas fases tiene objetivos específicos que contribuyen a la evolución del proyecto y su éxito.

- Fase de Concepción: El objetivo principal de esta fase es definir el alcance del proyecto, los objetivos principales y los riesgos. En esta etapa, se busca comprender las necesidades del cliente y establecer una visión clara del sistema que se va a desarrollar. "Si la fase inicial es larga, puede ser un indicio de una especificación inicial excesiva, lo que es contrario al espíritu del proceso unificado." (Acuña, 2020, párr.10). Por lo tanto, se debe evitar sobrecargar esta fase con demasiados detalles, enfocándose más en la planificación global y la identificación temprana de posibles obstáculos.

- Fase de Elaboración: El objetivo de esta fase es establecer la arquitectura base del sistema y refinar los requisitos del proyecto. "Durante la fase de elaboración, se espera que el equipo del proyecto capture una gran mayoría de los requisitos del sistema." (Acuña, 2020, párr.13). En este punto, se realiza un análisis profundo del sistema, identificando los componentes clave y definiendo cómo interactuarán entre sí. Es crucial porque cualquier error o falta de claridad en los requisitos en esta fase puede impactar negativamente en las siguientes etapas.
- Fase de Construcción: En esta fase, el objetivo es desarrollar el sistema completo o la mayor parte de él, con base en la arquitectura y los requisitos definidos en la fase de elaboración. "En esta fase, el resto del sistema se construye sobre la base establecida en la elaboración." (Acuña, 2020, párr.18). Se realizan iteraciones que permiten la entrega incremental de software funcional, lo que facilita la corrección temprana de problemas y el ajuste a cambios necesarios. El desarrollo de las funcionalidades principales tiene lugar en esta etapa.
- Fase de Transición: En la fase final, el objetivo es garantizar que el software esté listo para su implementación en el entorno de producción. "La fase de transición también incluye las conversiones del sistema y la capacitación de los usuarios." (Acuña, 2020, párr.23). Aquí se realizan pruebas finales, se corrigen errores, se migra la información necesaria y se entrena a los usuarios para asegurarse de que el sistema sea completamente funcional y cumpla con los objetivos esperados por el cliente.

Cada una de estas fases del Proceso Unificado juega un papel fundamental para garantizar el éxito del desarrollo del software, brindando un enfoque controlado y metódico para la gestión del ciclo de vida del proyecto.

6. La Metodología Ágil Extreme Programming (XP) se fundamenta en valores, prácticas y principios. De acuerdo con su criterio rescate 3 valores, 3 prácticas y 3 principios y justifique su respuesta en la escogencia en al menos 5 líneas de texto.

Extreme Programming (XP) se basa en una serie de valores, prácticas y principios que promueven el desarrollo ágil de software de alta calidad. Los valores, prácticas y principios seleccionados son los siguientes:

Valores:

1. **Comunicación:** XP promueve una constante interacción entre los miembros del equipo y los clientes, lo que permite una retroalimentación continua y evita malentendidos.
2. **Simplicidad:** Se enfoca en hacer solo lo necesario, eliminando complejidades innecesarias y manteniendo el código simple y manejable.
3. **Retroalimentación:** La evaluación constante del trabajo permite que los errores sean detectados y corregidos rápidamente, mejorando la calidad del producto final.

Prácticas:

1. **Desarrollo iterativo:** En XP, el desarrollo se realiza en pequeñas iteraciones, permitiendo que las funcionalidades sean ajustadas conforme a las necesidades cambiantes del cliente.
2. **Programación en pares:** Dos programadores trabajan juntos en una sola estación, lo que mejora la calidad del código al permitir una revisión continua.
3. **Integración continua:** El equipo fusiona los cambios en el código de manera frecuente para detectar errores en una etapa temprana y mantener el sistema siempre funcional.

Principios:

1. **Feedback rápido:** El principio del feedback rápido asegura que el equipo reciba y actúe sobre la retroalimentación lo antes posible para ajustar el desarrollo en la dirección correcta.
 2. **Simplicidad en el diseño:** XP fomenta el desarrollo de soluciones simples que puedan ser escaladas o ajustadas conforme se necesite, evitando diseños complejos desde el principio.
 3. **Coraje:** Los equipos deben ser valientes para hacer cambios necesarios, abordar problemas difíciles o eliminar características que no agreguen valor.
7. Para la Metodología Ágil Scrum, genere un cuadro donde cite y genere una breve descripción de los roles e indique quienes participan en cada rol.

A continuación, se presenta un cuadro con los roles de la Metodología Ágil Scrum, junto con una breve descripción y las personas que participan en cada uno de ellos:

Rol	Descripción	Participantes
Product Owner	Es el responsable de definir y priorizar las características del producto. Se asegura de que el equipo entienda las expectativas del cliente y se concentra en maximizar el valor del producto.	Representante del cliente, Gerente de producto
Scrum Master	Actúa como facilitador del proceso Scrum. Su función es eliminar impedimentos, asegurar que el equipo siga las prácticas de Scrum y fomentar la mejora continua.	Miembro del equipo ágil, Consultor externo
Team	Es el grupo de profesionales que trabajan en el desarrollo del producto. Este equipo es autoorganizado y multidisciplinario, lo que significa que posee todas las habilidades necesarias para entregar incrementos del producto.	Desarrolladores, Diseñadores, Ingenieros de calidad
Project Manager	Es el responsable de realizar la planificación, ejecución y cerrar el proyecto, además debe de velar que se cumplan las etapas y dar los recursos necesarios para llevar el proyecto, también debe de ser el comunicador entre los diferentes equipos de trabajos	Líder de proyecto, coordinador del proyecto, líderes de equipos.

Tabla 2 Cuadro Metodología Ágil Scrum

Conclusión

En conclusión, se ha aprendido que las metodologías pueden clasificarse en dos grandes categorías: las metodologías conducidas por los planes y las metodologías ágiles, cada una con sus características, ventajas y desventajas. Por ejemplo, se ha evidenciado que las metodologías ágiles, como Scrum y Extreme Programming, ofrecen mayor flexibilidad y adaptabilidad, mientras que las metodologías conducidas por planes son más adecuadas para proyectos con requisitos bien definidos.

Además, se ha concluido que el proceso unificado de desarrollo es fundamental para asegurar la calidad del software. A lo largo del análisis de sus fases, se ha identificado que la fase de elaboración es crucial, ya que en esta etapa se recopilan la mayoría de los requisitos del sistema, permitiendo así una construcción más sólida en las fases posteriores. Esto resalta la necesidad de dedicar el tiempo suficiente a cada fase del proceso, evitando especificaciones iniciales excesivas que puedan complicar el avance del proyecto, como lo señala Acuña (2020).

Asimismo, se ha aprendido que los valores, prácticas y principios de metodologías como Extreme Programming y Scrum ofrecen un marco robusto para el desarrollo ágil de software. La integración de estos valores en el equipo de trabajo fomenta un ambiente colaborativo que potencia la creatividad y la eficiencia. Por ejemplo, la práctica de la programación en parejas en Extreme Programming no solo mejora la calidad del código, sino que también facilita el aprendizaje mutuo entre los desarrolladores.

Finalmente, se ha reconocido que la formación continua y la adaptación son claves para el éxito en el desarrollo de software. La necesidad de mantenerse actualizado sobre las tendencias y mejores prácticas en la industria es esencial para enfrentar los constantes cambios y desafíos del entorno tecnológico. Así, la implementación de metodologías

adecuadas contribuye significativamente a maximizar la satisfacción del cliente y a garantizar la efectividad en la entrega de productos de software de alta calidad.

Referencias

Acuña, B. (2020). *Proceso unificado _ AcademiaLab*. Academia Lab. <https://academia-lab.com/enciclopedia/proceso-unificado/>

González Llobet, V. J. (2023, 20 de septiembre). *Las metodologías de desarrollo de software más populares - Donde las aplicaciones nacen*. Donde las aplicaciones nacen. <https://houseof.io/blog/las-metodologias-de-desarrollo-de-software-mas-populares/#:~:text=Las%20metodologías%20de%20desarrollo%20de%20software%20son%20conjuntos,planificación,%20el%20diseño,%20la%20implementación%20y%20las%20pruebas.>

Pantaleo, G., & Rinaudo, L. (2015). *Ingeniería de software*. Alfaomega Grupo Editor Argentino.

Power up your workflow: 8 metodologías de desarrollo de software para ser más eficiente. (2023, abril 29). *Talently Blog*. <https://talently.tech/blog/metodologias-desarrollo-software-workflow-eficiente/>

¿Qué son los proyectos de código abierto? (2021, diciembre 1). Arduino based PLC Controller for Industrial Applications by Industrial Shields. https://www.industrialshields.com/es_ES/blog/blog-industrial-open-source-1/que-son-los-proyectos-de-codigo-abierto-416

Valtx. (2023, abril 19). *Metodologías de desarrollo de software: ¿Qué son y para qué sirven?* Valtx. <https://www.valtx.pe/blog/metodologias-para-el-desarrollo-de-software-que-son-y-para-que-sirven>