Universidad Católica de El Salvador Centro Regional de Ilobasco.

Integrantes:

Castillo Bonilla Daniel Vinicio Carnet: 2023-CB-250

Carranza Villanueva Karla Lisseth Carnet: 2023-CV-251

Chavez Recinos Brian Josue Carnet: 2022-CR-211

Menjivar Ramos Carlos Javier Carnet: 2023-MR-251

Docente: Carlos Alfredo Hernández Quintanilla.

Asignatura: Sistemas Informáticos.

Tema: Metodologías Tradicionales y Metodologías Ágiles

Año: 2025.

ÍNDICE.

INTRODUCCIÓN. **3**
JUSTIFICACIÓN. **4**
OBJETIVOS. **5**
Objetivo General: 5
Objetivos específicos: 5
Descripción del proyecto elegido: **6**
¿Qué es el prototipado? **7**
Funciones del prototipado. 7
Definición de DevOps. **7**
Explicación sobre DevOps. 7
Metodología de DevOps. **8**
Principios básicos de DevOps: 8
Fases de la metodología Prototipado. **9**
Fases del Ciclo DevOps **10**
Preguntas de investigación **12**
¿Qué metodología fue más fácil de aplicar? 12
¿Cuál se adaptó mejor al caso de estudio? 12
¿Qué desafíos encontraron? 12
¿Qué metodología permitió una mayor flexibilidad ante cambios en los requisitos? 12
¿Cómo se manejaron los riesgos en cada enfoque? 13
¿Qué metodología fomentó más la colaboración entre el equipo? 13
CONCLUSIÓN. **14**

INTRODUCCIÓN.

En los primeros años del siglo XXI, la tecnología comenzó a expandirse a un ritmo acelerado, impulsando el desarrollo de software en distintos ámbitos. Sin embargo, los primeros sistemas informáticos eran costosos e ineficientes, lo que dificultaba su creación y mantenimiento. Para abordar estos desafíos, surgieron diversas metodologías de desarrollo de software, tanto tradicionales como ágiles, con el objetivo de optimizar los procesos, reducir costos y mejorar la calidad del producto final.

En este contexto, el uso de metodologías adecuadas se ha convertido en un pilar fundamental para el éxito de los proyectos de software. Mientras que las metodologías tradicionales ofrecen un enfoque estructurado y detallado, las metodologías ágiles han revolucionado la industria al priorizar la flexibilidad y la rápida adaptación a los cambios.

En este documento, analizaremos la aplicación de la metodología de **Prototipado y DevOps** en el desarrollo de un sistema de gestión hotelera, destacando sus ventajas y desafíos en comparación con ambos enfoques.

JUSTIFICACIÓN.

A partir de las metodologías Investigadas anteriormente, hemos descubierto la importancia de utilizar metodologías ágiles o tradicionales en nuestros proyectos para obtener un mejor rendimiento en el desarrollo de software, estas nos han permitido gestionar, agilizar y ordenar las fases de el desarrollo de un software

Aunque no sea fácil de implementar algunas metodologías es esencial comprender y efectuar el uso de estas, grandes empresas como google, amazon y meta. Implementan estas Metodologías ágiles para poder entregar software con tiempos de entrega cortos.

Por esto nos han interesado dos metodologías que nos parecen imperceptible, Metodología de Prototipado y Metodología DevOps, aunque se sabe que DevOps no es una metodología como tal sino más bien una corriente filosófica nos hemos interesado por esta, gracias a la complejidad que conlleva, utilizando contenedores, Integraciones continuas y despliegues continuas.

En cuanto a la metodología de prototipado: Esta nos llama la atención debido a su atractivo visual y el constante contacto con el cliente, para llevar a cabo el proyecto de forma que el cliente quede satisfecho con el exhaustivo proceso de elaboración de prototipos.

OBJETIVOS.

Objetivo General:

Desarrollar un sistema de gestión hotelera eficiente mediante la aplicación de metodologías de desarrollo seleccionadas, asegurando la correcta implementación y satisfacción de las necesidades del hotel.

Objetivos específicos:

- Implementar un sistema de gestión de reservas, empleados y clientes, incluyendo el procesamiento de pagos, utilizando una base de datos MySQL
- Analizar los beneficios que ofrece implementar Prototipado en el desarrollo de software, en comparación con los principios de DevOps y describir escenarios específicos donde cada enfoque sería más adecuado.
- Comparar las ventajas de DevOps frente al Prototipado en términos de capacidad de desarrollo, optimización de procesos y escalabilidad, analizando escenarios específicos para determinar el enfoque más adecuado en cada caso.

Descripción del proyecto elegido:

Nombre del proyecto: “Hotel Santa Clara”

El software planteado para el Hotel Santa Clara está dividido en módulos, lo que permite a la administración del hotel registrar clientes, empleados, crear o cancelar reservaciones y gestionar de manera más eficiente las habitaciones. Además, cuenta con un método mediante el cual el cliente puede optar por pagar en efectivo o con tarjeta. Para garantizar una mayor seguridad, se ha planteado incorporar un módulo específico que sólo permita el acceso a personal autorizado a dicha información.

Explicación técnicas.

Funcionalidad:

- Registro de clientes
- Registro de empleados
- Registro de habitaciones
- Reservaciones de habitaciones
- Gestión de pago
- Método de logueo

Tecnologías

- Lenguaje para desarrollar el software: Java
- Lenguaje de Base de Datos: MySql
- Framework para el diseño: JavaFX
- Compilador para el Software: IntelliJ IDEA Community Edition 2024.1.3
- Herramienta para diseño de Software: SceneBuilder
- Compilador para la Base de Datos: MySQL Workbench 8.0 CE

Metodologías Escogidas

- Prototipado
- DevOps

¿Qué es el prototipado?

El prototipado es una versión inicial de la idea de un producto o servicio. El prototipado nos permite probar, evaluar y validar si efectivamente la idea que tenemos en mente cumple los objetivos de la empresa y de los usuarios. Gracias a este prototipo podremos validar esas ideas que tenemos de productos o servicios pero que no sabemos cómo reflejarlas ni cómo llevarlas a un terreno palpable.

De hecho, otro de los puntos claves es poder testear con usuarios. Es decir, muchas veces, antes de pasar el proyecto al equipo de desarrollo para que comience con el código (que es una de las partes más caras de la producción de un producto digital), lo mejor es hacer un testing con usuarios. Por ejemplo, en un focus group, donde podremos probar este prototipo y ver qué opinan de él los usuarios. Ver dónde se atascan, las emociones a la hora de pulsar un botón u otro, etc.

Beneficios de prototipado

Uno de los mayores beneficios de los prototipos es, sin duda, la posibilidad de detectar de forma temprana los posibles problemas de diseño y fabricación. Además, la metodología de prototipado también nos puede ayudar a estimar las necesidades y los materiales que necesitaremos en un futuro, así como el tiempo de fabricación y el personal. Prototipar es perfecto si lo que quieres es:

- Presentar tu idea
- Diseñar la mejor versión para usuarios
- Conocer el alcance del proyecto

Gracias al testeo puedes resolver muchas cuestiones:

- **Minimizas riesgo**s, ya que gracias al testing y al prototipado te aseguras el éxito del producto, ya que has realizado varias pruebas antes y has podido comprobar que funciona y que las funcionalidades son las esperadas por el usuario.
- **Posibilidad de probar,** ya que vamos validando conceptos esenciales durante el desarrollo.
- **Mejora la experiencia de usuario,** previniendo fallos de diseño, principalmente de usabilidad.
- **Recibir feedback** de los clientes y usuarios finales.
- **Ayuda a lograr una ventaja competitiva,** ya que podemos incidir y trabajar en los aspectos que el consumidor percibe como importantes respecto a sus necesidades. O sea, desarrollamos un producto mejor adaptado y con más valor que el de la competencia.

- **La visión que nos da del producto** nos ayuda a hacernos una idea de los atributos que tendrá el producto. Esto nos ayudará a ir creando estrategias de diferenciación de cara a su comercialización.
- El descubrimiento temprano de los **problemas de diseño**.

Funciones del prototipado.

La fabricación de un ejemplar del proyecto cumple con ciertos objetivos que resultan de gran beneficio para empresas dedicadas a la comercialización de manufacturas. Entre algunas de las principales funciones que ofrece esta técnica se encuentran las siguientes:

- Identificar las características del producto final. El diseño de modelos de productos brinda una perspectiva general de las cualidades físicas y funcionales del artículo en cuestión. El equipo encargado analiza estos aspectos y determina qué propiedades deben cambiarse, eliminarse o ajustarse para un resultado final más satisfactorio. Además, se busca alinear los intereses comerciales con la necesidad del público consumidor.
- Definir parámetros de producción. Con esto nos referimos al proceso en donde se estudian la cantidad de materiales que deberán emplearse para traer el artículo al mundo físico o digital. A través de ello, la empresa puede entender la clase de herramientas que deben usarse durante la producción, la cantidad de equipo humano requerido, etc. También funciona para determinar si es necesario capacitar al personal para este nuevo proyecto.

Definición de DevOps.

DevOps es una combinación de desarrollo (dev) de software y operaciones (ops). Se define como una metodología de ingeniería de software que tiene como objetivo integrar el trabajo de los equipos de desarrollo y los equipos de operaciones al facilitar una cultura de colaboración y responsabilidad compartida.

Explicación sobre DevOps.

DevOps puede explicarse mejor como personas que trabajan juntas para concebir, crear y entregar software seguro a la máxima velocidad. Las prácticas de DevOps permiten a los equipos de desarrollo (dev) y operaciones (ops) de software acelerar la entrega a través de la automatización, la colaboración, los comentarios rápidos y la mejora iterativa. Partiendo de un enfoque ágil para el

desarrollo de software, un proceso de DevOps amplía el enfoque interdisciplinario de la creación y el envío de aplicaciones de una manera más rápida e iterativa.

Al adoptar un proceso de desarrollo de DevOps, está tomando la decisión de mejorar el flujo y la entrega de valor de su aplicación fomentando un entorno más colaborativo en todas las etapas del ciclo de desarrollo. DevOps representa un cambio de mentalidad para la cultura de TI. DevOps se centra en el desarrollo incremental y la entrega rápida de software ya que se basa en las prácticas ágiles, Lean y la teoría de sistemas. El éxito se basa en la capacidad de crear una cultura de responsabilidad, colaboración mejorada, empatía y responsabilidad conjunta por los resultados comerciales.

Metodología de DevOps.

La metodología DevOps tiene como objetivo acortar el ciclo de vida del desarrollo de sistemas y proporcionar una entrega continua con alta calidad de software. Enfatiza la colaboración, la automatización, la integración y los ciclos rápidos de comentarios. Estas características ayudan a garantizar una cultura de creación, prueba y lanzamiento de software que sea más confiable y a mayor velocidad. Esta metodología comprende cuatro principios clave que guían la efectividad y eficiencia del desarrollo e implementación de aplicaciones. Estos principios, que se enumeran a continuación, se centran en los mejores aspectos del desarrollo de software moderno.

Principios básicos de DevOps:

1. **Automatización del ciclo de desarrollo del software.** Esto incluye la automatización de pruebas, compilaciones, lanzamientos, el aprovisionamiento de entornos de desarrollo y otras tareas manuales que pueden ralentizar o introducir errores humanos en el proceso de entrega de software.
2. **Colaboración y comunicación.** Un buen equipo de DevOps tiene automatización, pero un gran equipo de DevOps también tiene una colaboración y comunicación efectivas.
3. **Mejora continua y minimización de desperdicio.** Desde la automatización de tareas repetitivas hasta la observación de las métricas de rendimiento para encontrar formas de reducir los tiempos de lanzamiento o el tiempo medio de recuperación, los equipos de DevOps de alto rendimiento buscan de manera periódica áreas que podrían mejorarse.
4. **Hiper enfoque en las necesidades del usuario con breves ciclos de comentarios.** A través de la automatización, la mejora de la comunicación y colaboración, y la mejora

continua, los equipos de DevOps pueden tomarse un momento y centrarse en lo que realmente quieren los usuarios y en cómo ofrecérselo.

Fases de la metodología Prototipado.

1. **Identificación de requisitos iniciales:**
    a. Se recopilan los requisitos básicos del sistema, aunque no estén completamente definidos.
    b. Se identifican las áreas de mayor incertidumbre o complejidad que necesitan ser validadas.
2. **Diseño rápido del prototipo:**
    a. Se crea un prototipo rápido y funcional que cubre las funcionalidades clave o las áreas de mayor incertidumbre.
    b. El prototipo puede ser de baja fidelidad (esbozos, wireframes) o de alta fidelidad (interactivo y cercano al producto final).
3. **Evaluación del prototipo:**
    a. El prototipo se presenta a los usuarios o stakeholders para que lo prueben y proporcionen retroalimentación.
    b. Se identifican problemas, mejoras y requisitos adicionales.
4. **Refinamiento y mejora:**
    a. Con base en la retroalimentación, el prototipo se mejora y se ajusta.
    b. Este proceso puede repetirse varias veces hasta que los requisitos estén claros y el diseño sea satisfactorio.
5. **Desarrollo del producto final:**
    a. Una vez que el prototipo ha sido validado, se procede al desarrollo del producto final utilizando los requisitos refinados.

Fases del Ciclo DevOps

Aunque DevOps es más un conjunto de prácticas y una cultura que una metodología con fases rígidas, se pueden identificar etapas clave en su implementación:

1. **Planificación:**
    - Definición de requerimientos y objetivos.
    - Selección de herramientas y establecimiento de KPIs (indicadores clave de rendimiento).
2. **Desarrollo e Integración Continua (CI):**
    - Los desarrolladores integran continuamente su código en un repositorio compartido.
    - Se ejecutan pruebas automatizadas para detectar errores de manera temprana.
3. **Entrega Continua (CD):**
    - El código que ha pasado todas las pruebas se despliega en entornos de staging o pre-producción.
    - Se prepara el pipeline de despliegue para llevar el código a producción de forma automática.
4. **Despliegue:**
    - El código se implementa en producción.
    - Se utilizan herramientas de orquestación y contenedores para asegurar la consistencia y escalabilidad.
5. **Operaciones y Monitoreo:**
    - Supervisión en tiempo real del rendimiento y la estabilidad del sistema.
    - Registro y análisis de incidentes para realizar mejoras continuas.
6. **Retroalimentación y Mejora Continua:**
    - Se recogen datos y comentarios de usuarios y equipos.
    - Se realizan ajustes en el código, la infraestructura y los procesos para optimizar el ciclo de vida del software.

Roles y responsabilidades para la metodología DevOps:

En DevOps, aunque no existe una estructura jerárquica tan rígida como en otros modelos tradicionales, sí se identifican roles y responsabilidades que ayudan a alinear los objetivos de todos los equipos. Algunos roles típicos son:

1. I**ngeniero DevOps:**
   Se encarga de la automatización, configuración de pipelines CI/CD, orquestación de contenedores (por ejemplo, Docker, Kubernetes) y monitoreo de aplicaciones en producción.

2. **Desarrollador:**
   Colabora en la creación del software y se involucra en la integración y despliegue continuo, asegurándose de que el código cumpla con los estándares necesarios para integrarse en el pipeline.

3. **Ingeniero de Operaciones:**
   Se centra en la infraestructura, la administración de servidores y la gestión de la escalabilidad y disponibilidad de la aplicación.

4. **Especialista en QA (Quality Assurance):**
   Automatiza pruebas y participa en la validación continua del software a lo largo del ciclo de vida.

5. **Especialista en Seguridad (DevSecOps):**
   Integra las prácticas de seguridad desde el inicio del desarrollo, evaluando vulnerabilidades y asegurando el cumplimiento de normativas.

Preguntas de investigación

¿Qué metodología fue más fácil de aplicar?

Sin duda, la metodología de Prototipado ha sido la más sencilla de implementar debido a su baja complejidad y estructura clara. Gracias al análisis previo de requerimientos, facilita la orientación del usuario en la definición de sus necesidades, permitiéndole visualizar diferentes opciones mediante prototipos. Aunque el proceso inicial puede ser tedioso debido a la iteración de

prototipos desechables, una vez aprobado el diseño, la metodología resulta muy eficiente y cómoda para el desarrollo.

¿Cuál se adaptó mejor al caso de estudio?

La metodología que mejor se adaptó al proyecto fue Prototipado. Dado que el cliente no tenía una idea completamente definida sobre todas las funcionalidades del sistema, este enfoque permitió visualizar el software en etapas tempranas. Gracias a la retroalimentación continua, se realizaron ajustes y mejoras en cada iteración, lo que permitió desarrollar un sistema alineado con las necesidades reales del hotel. Esto no solo optimizó el proceso de desarrollo, sino que también redujo costos asociados a cambios tardíos y mejoró la satisfacción del cliente.

¿Qué desafíos encontraron?

Tabla 1: *Desafíos encontrados en las metodologías de Prototipado y Devops.*

Prototipado	DevOps
● Costo y tiempo en la creación de prototipos desechables. ● Dificultad para garantizar una fidelidad del 100% al prototipo aprobado. ● Posible insatisfacción del cliente por diferencias entre prototipo y producto final	● Complejidad en el uso y gestión de contenedores. ● Retos en la implementación de medidas de seguridad efectivas. ● Desafíos en la automatización y orquestación de procesos.

Nota: Esta es una representación de los desafíos para las metodologías Prototipado y DevOps

¿Qué metodología permitió una mayor flexibilidad ante cambios en los requisitos?

DevOps permitió implementar modificaciones de manera rápida y eficiente, garantizando la estabilidad del sistema mediante pruebas automatizadas y monitoreo continuo. Su capacidad de automatización facilita la adaptación a nuevas necesidades del cliente sin interrumpir el flujo de trabajo, asegurando así entregas más ágiles y seguras.

¿Cómo se manejaron los riesgos en cada enfoque?

Tabla 2: *Manejo de riesgos de cada enfoque*

Prototipado	DevOps
● Enfocar el diseño en las necesidades y preferencias del cliente. ● Iteraciones constantes para ajustar el diseño hasta lograr un producto satisfactorio. ● Comunicación clara con el cliente sobre la naturaleza iterativa del proceso.	● Capacitación en el uso de herramientas como Docker y Kubernetes. ● Implementación de pruebas de seguridad en cada iteración. ● Configuración y optimización de pipelines de CI/CD para minimizar errores.

Nota: Esta es una representación del manejo de riesgos

¿Qué metodología fomentó más la colaboración entre el equipo?

Ambas metodologías fomentaron la colaboración dentro del equipo, pero en diferentes aspectos del desarrollo. **Prototipado** promovió la interacción constante con el cliente y entre los desarrolladores, asegurando que las funcionalidades del sistema se ajustaran a sus necesidades mediante la retroalimentación continua. Por otro lado, **DevOps** fortaleció la colaboración entre los equipos de desarrollo y operaciones, facilitando la integración y entrega continua del software.

Sin embargo, si se considera la comunicación interna del equipo durante todo el ciclo de desarrollo, **DevOps** fue la metodología que más impulsó la colaboración, ya que requiere una coordinación constante entre desarrolladores, testers y administradores de sistemas para garantizar implementaciones rápidas y eficientes.

CONCLUSIÓN.

Tras analizar las metodologías de Prototipado y DevOps, podemos concluir que ambos enfoques ofrecen ventajas distintivas para el desarrollo de software, aunque sirven diferentes propósitos y contextos. El Prototipado resulta especialmente valioso en proyectos como el sistema de gestión hotelera "Hotel Santa Clara", donde la validación temprana con usuarios y la iteración del diseño son cruciales para asegurar que el producto final cumpla con las expectativas del cliente.

Por otro lado, DevOps, aunque no es estrictamente una metodología sino una cultura de desarrollo, proporciona un marco robusto para la entrega continua y la integración eficiente entre desarrollo y operaciones. Su enfoque en la automatización, colaboración y mejora continua lo hace particularmente efectivo para proyectos que requieren despliegues frecuentes y alta escalabilidad.

La combinación de ambos enfoques puede resultar beneficiosa: mientras el Prototipado nos permite validar rápidamente los requisitos y la experiencia del usuario, DevOps nos proporciona las herramientas y prácticas necesarias para llevar estos prototipos a producción de manera eficiente y mantenerlos de forma sostenible.

Para el proyecto del Hotel Santa Clara, esta combinación permite desarrollar un sistema robusto y centrado en el usuario, con la capacidad de evolucionar y escalar según las necesidades del negocio, mientras se mantiene un alto nivel de calidad y eficiencia en el desarrollo y despliegue del software.

BIBLIOGRAFÍA

Bennett, L. (2024, August 13). *Modelo de prototipo en ingeniería de software*. Guru99. https://www.guru99.com/es/software-engineering-prototyping-model.html

de Expertos en Empresa, E. (2023, February 20). Prototipado: qué es y cómo funciona. *VIU Colombia*. https://www.universidadviu.com/co/actualidad/nuestros-expertos/prototipado-que-es-y-como-funciona

¿Qué es DevOps? (2022, February 10). Gitlab.com; GitLab. https://about.gitlab.com/es/topics/devops/

Bello, E. (2023, 29 junio). ¿Qué es el prototipado y cómo prototipar un producto? *Thinking for Innovation*. https://www.iebschool.com/blog/que-es-el-prototipado-digital-business/#:~:text=El%20prototipado%20es%20una%20versi%C3%B3n,empresa%20y%20de%20los%20usuarios.