

Ingeniería de Software: Parcial

Edward Javier Parada Silva - 2182070
Daniel David Delgado Cervantes - 2182066

1. Aseguramiento De La Salud

1.1. - 1.4. [Ver anexos]

1.2. Documentación de historias de usuario

- Como afiliado, debo poder solicitar un servicio mediante una remisión válida y autorizada por la EPS para así continuar con el proceso médico.
- Como auditor de la EPS debo poder asignar una glosa a una factura con ítems individuales que correspondan con lo definido en la factura, así mismo, debo poder realizar los ajustes necesarios dependiendo de las negociaciones entre la IPS y la EPS.

1.5. Proponga 3 requerimientos no funcionales que pueden ser aplicados al caso de estudio

- El sistema debe ser accesible desde el navegador web Google Chrome versión 96
- El sistema no puede caerse por más de 3 minutos al día
- El sistema debe implementarse con OracleSQL para las bases de datos, Springboot para backend y Angular para frontend

1.6. Plantee por lo menos 5 preguntas que le ayudarían a entender mejor la situación descrita en el enunciado

1. ¿Existe un máximo tiempo de regateo de una glosa?
2. Si no existe IPS capaz de cubrir todos los servicios solicitados, ¿Es posible dividir los procedimientos realizados entre varias?
3. ¿Qué administrativos son los encargados realizar la autorización de la remisión?
4. ¿Qué procedimiento debe seguir un paciente en el caso de que una remisión no sea autorizada?
5. ¿Puede un paciente tener más de una remisión activa al mismo tiempo?

2. Compartamos El Vehículo

2.1. - 2.4. [Ver anexos]

2.2. Documentación de historias de usuario

- Como conductor de la aplicación, requiero poder publicar mis rutas a la aplicación para que así los demás usuarios puedan solicitar un cupo dentro de la ruta acordada.
- Como administrador de la aplicación, requiero poder ver las quejas que los usuarios reciban para así poder suspender a los malos conductores y pasajeros.

2.5. Proponga 3 requerimientos no funcionales que pueden ser aplicados al caso de estudio

- El sistema debe estar disponible para Android e IOS
- El sistema debe correr en celulares de bajo rendimiento
- El sistema no puede tener un costo base menor a 20 millones de pesos

2.6. Plantee por lo menos 5 preguntas que le ayudarían a entender mejor la situación descrita en el enunciado

1. ¿Qué pasa si un conductor, durante un trayecto, se accidenta?
2. ¿Qué puede llevar a la suspensión de un pasajero?
3. ¿Qué información deberían poder ver los conductores sobre los pasajeros?
4. ¿Tras cuantos incumplimientos se debería suspender a un usuario?
5. ¿Qué herramientas de seguridad deben estar disponibles para el pasajero?

3. Modelos de procesos software

3.1. ¿Cuál es el objetivo central de un modelo de procesos de desarrollo de software?

Los modelos de procesos de desarrollo de software tienen como objetivo principal el organizar las diferentes etapas del desarrollo. Este orden, propuesto por cada una de las metodologías, tiene la intención de reducir los casos de re-proceso o futuros cambios dentro del desarrollo del proyecto.

3.2. ¿Cuales modelos de proceso de desarrollo software conoce y que características tienen? Realice un cuadro comparativo de los modelos.

3.2.1. Modelos conocidos

Modelo Cascada

El modelo de cascada, o el modelo clásico, es una manera de organizar las etapas de desarrollo a manera de línea, una tras la otra. Así mismo, y debido a esta composición, se obtiene un modelo poco flexible en el cual en el caso de realizar cambios en términos de diseño o estructura, se torna problemático debido a la misma metodología que no lo permite.

Modelo Espiral o Cíclico

El modelo de espiral puede verse como una extensión de la metodología de cascada. Esta, en esencia, realiza de manera cíclica el modelo de cascada. De esta manera, permite la realización de las diferentes etapas múltiples veces durante el ciclo de desarrollo, facilitando así un poco el manejo de los cambios y errores. Aún así, sigue siendo rígida puesto que no permite realizar dichos cambios sino hasta el inicio de un nuevo ciclo.

Modelo De Prototipos

El modelo de prototipos es una metodología basada en la retroalimentación recibida a partir del desarrollo de prototipos de bajo costo. Es a partir de estos prototipos, realizados idealmente en un corto tiempo, y la retroalimentación recibida tras la presentación y evaluación al cliente, se realiza el levantamiento de requerimientos con los cuales se realizaría la versión final del software.

SCRUM

La metodología SCRUM, que hace parte de las metodologías de desarrollo ágiles, se basa en diferentes técnicas en las cuales impulsan el trabajo en equipo y de manera colaborativa. Así mismo, la metodología SCRUM, trabaja a partir de entregas parciales y regulares del producto. De esta manera, se busca mantener una alta flexibilidad ante los entornos constantemente cambiantes y de alta complejidad.

3.2.2. Cuadro comparativo

	Modelo Cascada	Modelo Espiral o Cíclico	Modelo De Prototipos	SCRUM
Flexibilidad	Baja	Media	Media	Alta
Velocidad De Desarrollo	Alta	Media	Alta	Media
Casos de uso	Proyectos con requerimientos claros y bien definidos	Proyectos con requisitos poco claros o complejos	Proyectos con requisitos poco claros y complejos	Proyectos de gran complejidad y con requisitos poco claros
Limitaciones	Se organiza de manera que no se puede regresar a fases anteriores, muy rigida	No es facil de implementar, pues es costoso y depende del análisis de riesgos realizado	Media escala	Gran Escala
Escala	Baja escala	Media/Gran escala	Media escala	Gran Escala

3.3. ¿Cuáles son las actividades que están involucradas en el proceso de desarrollo software, con cuál de ellos se siente más afín y por qué?

Las actividades involucradas dentro del desarrollo de software pueden definirse como:

- **Fase de Determinación de requerimientos:** En esta fase es donde se determinan las necesidades del cliente en cuanto al software que desea adquirir. Es en esta parte en donde se define qué partes son necesarias desarrollar y cuales no son importantes para las necesidades del cliente. Es de gran importancia que esta se realice de la mejor manera posible puesto que es en esta fase en la cual se siembran las semillas para el desarrollo completo del proyecto.
- **Fase de Diseño:** En la fase de diseño, es donde se realiza, a partir de lo establecido en la determinación de los requerimientos, se define el como será el funcionamiento de cada una de las partes las cuales compondrán la solución a desarrollar. Así mismo, es en esta fase que se definen el como se verá la solución, al igual que el como esta estará compuesta.
- **Fase de Implementación:** La fase de implementación consta de como tal el desarrollo del software, siendo así, es en esta fase en la cual se trabaja en pro de la realización del proyecto. Esta fase compete la mayor parte del desarrollo y es en la cual se realiza la creaión del producto final.

- **Fase de Verificación** La fase de verificación consta de la verificación de lo desarrollado con el cliente. Es aquí donde se espera el presentar el producto desarrollado que de solución a la problemática a la cual el cliente se estaba enfrentando.
- **Fase de Mantenimiento** La fase de mantenimiento es la fase en la cual la solución se encuentra ya como tal funcionando. Es en esta fase en la cual se realiza la corrección de errores básicos al igual que ajustes al software desarrollado con el fin de que este aún sea útil para el cliente.

En cuanto a que actividad es la cual es más a fin a nosotros, nos identificamos con la fase de implementación. Esto se debe principalmente a que, de manera general y por experiencia, esta es la fase en la cual como estudiantes nos hemos desarrollado mayoritariamente. Así mismo, puede considerarse que esta fase es la más activa en cuanto es donde se plantea una gran cantidad de resolución de problemas.

3.4. Identifique y describa un ejemplo donde utilizaría un modelo de proceso ágil (indique cual) y otro donde utilizaría un método tradicional (indique cual)

Un caso en el cual se utilizaría una metodología ágil, en este caso SCRUM, sería en el desarrollo de un proyecto en el cual se esté realizando la reestructuración de los sistemas de información de una universidad. En este caso, debido a la gran escala y gran cantidad de partes que implican la solución, el emplear una metodología ágil es lo más racional en términos de costo y tiempo.

Así mismo, el caso en el cual se emplearía un método tradicional, en este caso de cascada, sería en un proyecto personal de baja escala como puede ser el desarrollo de una aplicación web que retorne la información de los resultados de los últimos partidos de los equipos de la Universidad. Esto se debe a que el alcance es relativamente bajo y es posible realizarla con esta metodología.

4. Ingeniero de Software

4.1. ¿Cuál cree que es el rol que juega un ingeniero de software en la industria de tecnologías de la información?

El principal rol del ingeniero de software está en la parte del diseño y el análisis. Es decir, el enfoque principal de ellos está en realizar el análisis, especialmente a un nivel conceptual, de los proyectos con el fin de definir las bases con las que se realizarán el resto de las actividades del desarrollo.

4.2. ¿Cuáles considera que deben ser las habilidades requeridas por un ingeniero de software para desempeñarse actualmente y como cree que pueden ser fortalecidas en el curso de ingeniería de software?

La principal habilidad de un ingeniero de software está en la abstracción. Es a partir de esta abstracción de las necesidades del cliente son llevadas hacia el campo de trabajo de los desarrolladores. Siendo así, el tener una gran capacidad abstraer los diferentes conceptos que se trabajan en la vida real, es de gran importancia. Así mismo, se espera que el ingeniero de software tenga la habilidad de tomar decisiones de diseño las cuales faciliten el trabajo de los desarrolladores sin despegarse de las necesidades y requerimientos del cliente.

Ya en cuanto a los efectos que puede tener el curso de ingeniería de software, se espera que estas puedan fortalecerse en un nivel técnico-práctico el cual permita sus estudiantes el mejorar sus capacidades de abstracción, facilitando la traducción de la realidad a nivel de software; y su capacidad de tomar inteligentes

decisiones de diseño que les permitan sacar adelante los diferentes retos y proyectos a los cuales puedan enfrentarse.