ASPECTOS CONCEPTUALES

1. Enumere y describa los pasos implicados dentro de la técnica (Articulo) “De los Procesos de Negocio a los Casos de Uso” [30]
2. Identificar los procesos de negocio. El primer paso es identificar los procesos de negocio relevantes para el sistema en cuestión. Los procesos de negocio representan las actividades y flujos de trabajo clave que ocurren dentro de una organización.
3. Definir los actores. Los actores son las entidades externas que interactúan con el sistema. Se deben identificar y describir los diferentes actores que participan en los procesos de negocio.
4. Descomponer los procesos de negocio. Una vez identificados los procesos de negocio, se deben descomponer en actividades más pequeñas y manejables.
5. Identificar los casos de usos. Con la descomposición de los procesos de negocio, se pueden identificar los casos de uso. Éstos representan las interacciones entre los actores y el sistema para lograr un objetivo específico.
6. Documentar los casos de usos. Se debe documentar cada caso de uso identificado. Esto implica describir detalladamente el flujo de eventos que ocurren entre el actor y el sistema para lograr el objetivo del caso de uso. También se pueden incluir precondiciones y postcondiciones.
7. Relacionar los casos de uso con los procesos de negocio. Por último, se debe establecer una relación clara entre los casos de uso y los procesos de negocio. Esto se realiza mediante la identificación de qué casos de uso están asociados a cada actividad descompuesta en los procesos de negocio.
8. Explique para qué se utiliza el patrón Adaptador. (Explicación y Diagrama de Clases)

El patrón ADAPTADOR es utilizado en el diseño de software para permitir que dos interfaces incompatibles trabajen juntas. Su objetivo principal es convertir la interfaz de una clase en otra interfaz esperada por el cliente, de manera que puedan colaborar sin problemas.

El patrón Adaptador se utiliza cuando se tienen las siguientes situaciones:

* Se desea utilizar una clase existente que proporciona una funcionalidad útil, pero su interfaz no coincide con la interfaz requerida por el cliente.
* Se necesitan reutilizar clases existentes que no son compatibles con el resto del sistema debido a diferencias de interfaz.
* Se desea crear una clase reutilizable que coopere con clases no relacionadas, es decir, se desea lograr una adaptación entre clases independientes.

Diagrama

Descripción generada automáticamente

1. Explique las siguientes expresiones en OCL:

* Context Proyecto inv: self.participa -> select (edad > 50) ->notEmpty() [5].

La siguiente expresión OCL establece una restricción en el contexto de la clase ‘Proyecto’ que verifica si hay al menos 5 elementos en la asociación “participa” cuya edad sea mayor que 50.

* Context Job inv: self.empleado.edad > 21 [5].

La siguiente expression OCL establece una restricción en el contexto de la clase ‘Job’ que verifica si al menos 5 objetos “Job” tienen un empleado con una edad mayor a 21.

1. Explique cómo se realiza paso a paso la transformación de un Diagrama de Clases de un Diagrama Entidad Relación
2. Identificar las entidades: Se deben identificar las clases que representan entidades en el sistema. Cada clase se convierte en una entidad en el DER. Se crean nuevos atributos identificadores para cada entidad.
3. Mapear atributos: Para cada entidad identificada previamente, se deben mapear los atributos de la clase a los atributos de la entidad correspondientes en el DER teniendo en cuenta los tipos de datos.
4. Identificar las relaciones: Las asociaciones se transforman en interrelaciones. Además se mantiene, en el modelo de datos, la misma multiplicidad de la asociación.
5. Mapear claves primarias y foráneas: Se debe identificar las claves primarias en el Diagrama de Clases y mapear estas claves como primarias en el DER. En caso de existir asociaciones entre las entidades, se deben mapear las claves primarias de una entidad como claves foráneas en la otra entidad para establecer la relación.
6. Explique el siguiente diagrama que se relaciona con las metodologías Ágiles

**Diagrama

Descripción generada automáticamente**

Scrum es un proceso en el que se aplican regularmente un conjunto de buenas prácticas para trabajar colaborativamente y obtener el mejor resultado posible de un proyecto.

El proceso comienza con la elaboración del requisito futuro o mejor llamado ´Product Backlog. Se trata de un documento genérico que recoge el conjunto de tareas, los requerimientos y las funcionalidades requeridas por el proyecto. Cualquier miembro puede modificar este documento pero el ‘Product Owner’ es el único con la jerarquía para establecer prioridades.

La segunda etapa comienza con la definición del ‘Sprint Backlog’ o tareas del sprint. En resumidas palabras, es un documento que recoge las tareas a realizar y quién las desempeña. El Sprint es el periodo en el que se realizan todas las acciones pactadas con el Sprint Backlog, que supone entregas parciales para ir testeando el producto final.

Este ciclo deberá repetirse hasta que todos los elementos del Backlog hayan sido entregados. No se deben dejar tiempos sin productividad entre los distintos Sprints.

Luego, todas las acciones que se realizan deben tener un control. En esta fase final del Sprint, se revisa todo el trabajo para tener feedback sobre el desarrollo del producto. Puede ser una reunión informal mientras el objetivo del ‘Sprint Review’ sea brindar la mayor transparencia posible tanto al equipo como al cliente.

Suele tener una duración de 4 horas para Sprints de 4 semanas. La persona encargada de realizar este trabajo es el Product Owner. En los Sprints Reviews, se trata de explicar los ítems del Product Backlog y asegurarse de que hayan sido finalizados. Al mismo tiempo, el equipo de desarrollo se encarga de hacer la demostración del incremento terminado durante el Sprint.

A partir de ahí, se hace una review del proyecto y se tratan los siguientes pasos a seguir. En caso de haber una respuesta del cliente, el Product Owner reorganizaría el Product Backlog.