Celeste Obtenido de la práctica

Verde Obtenido de la Bibliografía

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Cuestionario Final   1. ¿Cuáles son las principales características de la POO? DESCRIBIR brevemente. 30 lineas  |  | | --- | | Abstracción: Es la definición conceptual de un conjunto de objetos identificados en el dominio.  Las demás características están descriptas en la práctica pero sin explicación.  **Encapsulamiento**: los atributos privados implican que serán visibles sólo para los objetos externos a través de los métodos de la clase, una técnica que se conoce como encapsulamiento u ocultamiento de la información. (Se refiere a los atributos)  **Ocultamiento**: El ocultamiento de información significa que los métodos de los objetos deben estar disponibles para otras clases, por lo que comúnmente los métodos son públicos, lo cual significa que se pueden invocar desde otras clases. (Se refiere a los métodos)  Hacer los atributos privados implica que serán visibles sólo para los objetos externos a través de los métodos de la clase, una técnica que se conoce como encapsulamiento u ocultamiento de la información  **Herencia**: varias clases pueden tener los mismos atributos y/o métodos. Cuando esto ocurre, se crea una clase general que contiene los atributos y métodos comunes. La clase especializada recibe o hereda los atributos y métodos de la clase general. Además, la clase especializada tiene atributos y métodos únicos que sólo se definen en ella. Al crear clases generalizadas y permitir que la clase especializada herede los atributos y métodos ayudamos a fomentar la reutilización, ya que el código se utiliza muchas veces. Esto también ayuda a mantener el código existente del programa. Así, el analista puede definir los atributos y métodos una vez, pero usarlos muchas veces en cada clase heredada.  **Polimorfismo**: Una subclase hereda un método padre pero le puede agregar elementos o modificarlo. La subclase puede cambiar el tipo de los datos, o puede cambiar la forma en que trabaja el método. Por ejemplo, podría haber un cliente que reciba un descuento adicional por volumen, con lo que se usaría un método modificado para calcular el total del pedido. Se dice que el método de la subclase redefine al método de la superclase.  Comportamientos alternativos entre las clases derivadas en las metodologías orientadas a objetos.  Cuando varias clases heredan tanto atributos como comportamientos, el comportamiento de una clase derivada podría ser distinto al de su clase base o al de sus clases derivadas hermanas. |  1. ¿Qué aprendiste sobre el paradigma orientado a objetos? 30 lineas  |  | | --- | | Es una forma de pensar, diseñar, construir e implementar programas basados en objetos que combinan estados, comportamientos e identidades.   * El estado está compuesto por datos, * el comportamiento es una serie de acciones que se realizan y son llamados métodos y * la identidad es una identidad del objeto que lo diferencia del resto de los objetos. |  1. ¿Qué es UML?. Desarrolla una explicación para que cualquier persona lo entienda y sus principales componentes. 30 lineas  |  | | --- | | UML es un Lenguaje de Modelado Unificado basado en una notación gráfica la cual permite: especificar, construir, visualizar y documentar los objetos de un sistema programado.  No es una metodología  **Componentes:** Diagrama de clases. Diagrama de objetos. Diagrama de actividades. Diagrama de secuencia. Diagrama de estado. Diagrama de casos de uso.  Los principales componentes del UML son cosas, relaciones y diagramas. Los diagramas están relacionados entre sí. Las cosas estructurales son más comunes; incluyen clases, interfa ces, casos de uso y muchos otros elementos que proveen la forma de crear modelos. Las cosas estructurales permiten al usuario describir relaciones. Las cosas de comportamiento describen la forma en que trabajan las cosas. Las cosas de agrupamiento se utilizan para definir límites. Las cosas de anotaciones permiten al analista agregar notas a los diagramas.  UML es el acrónimo de "Lenguaje Unificado de Modelado" en inglés, que es un lenguaje de modelado visual utilizado en la programación orientada a objetos para representar y diseñar sistemas de software. UML proporciona un conjunto estandarizado de herramientas para documentar el análisis y diseño de un sistema de software. El conjunto de herramientas de UML incluye diagramas que permiten a las personas visualizar la construcción de un sistema orientado a objetos, algo similar a la forma en que los planos de construcción permiten a las personas visualizar la construcción de un edificio. UML se basa en una técnica orientada a objetos conocida como modelado de casos de uso, que describe qué hace el sistema sin describir cómo lo hace. 36 y 6. |  1. ¿Cuál es la diferencia entre un caso de uso CU y un diagrama de casos de uso? 20 lineas  |  | | --- | | **Caso de uso**: “Un caso de uso especifica un conjunto de secuencias de acciones, incluyendo variantes, que el sistema puede ejecutar y que produce un resultado observable de valor para un actor particular”. El caso de uso es una herramienta textual.  **Diagrama de CU**: Es una herramienta de modelado gráfica. En los diagramas de casos de uso se muestran: casos de uso (representados en forma de elipses), actores (en forma de personajes) y relaciones (en forma de líneas y/o flechas).  Diagrama de CU: Un diagrama de casos de uso, que describe la forma en que se utiliza el sistema.  **Diferencia**: El diagrama de CU da una vista general de cómo se utiliza un sistema, un CU especifica o describe una funcionalidad del sistema.  Un caso de uso (CU) es una descripción detallada de una interacción específica entre un usuario y un sistema, que describe qué hace el sistema sin describir cómo lo hace. Un diagrama de casos de uso, por otro lado, es una representación visual de los casos de uso de un sistema y cómo se relacionan entre sí. El diagrama de casos de uso muestra los actores que interactúan con el sistema y los casos de uso que realizan las acciones desencadenadas por los eventos iniciados por los actores. En resumen, un caso de uso es una descripción detallada de una interacción específica, mientras que un diagrama de casos de uso es una representación visual de los casos de uso de un sistema y cómo se relacionan entre sí. Página 7 |  1. Dentro de los elementos de un caso de uso, ¿Qué representa un actor? 10 lineas      |  | | --- | | **Actores**: son los roles que pueden tomar las distintas persona y sistema que interactúan con el sistema.  Los actores se utilizan para identificar quiénes son los usuarios del sistema y qué roles desempeñan en las interacciones con el sistema. 7 |  1. ¿Cuáles son los bloques principales de UML? 30 lineas  |  | | --- | | **Diagramas**: diagrama de clases, diagrama de casos de usos, diagramas de interacción, etc.  **Elementos del modelo**: clases, interfaces, componentes, casos de usos, etc.  **Relaciones**: asociación, generalización, dependencias, etc.  Estas cosas consisten en notas que se pueden unir a cualquier cosa en UML: objetos, comportamientos, relaciones, diagramas o cualquier cosa que requiera descripciones detalladas, suposiciones o cualquier información relevante para el diseño y la funcionalidad del sistema.  Los bloques principales de UML son cosas, relaciones y diagramas. Las cosas estructurales son más comunes e incluyen clases, interfaces, casos de uso y muchos otros elementos que proveen la forma de crear modelos. Las cosas estructurales permiten al usuario describir relaciones. Las cosas de comportamiento describen la forma en que trabajan las cosas. Las cosas de agrupamiento se utilizan para definir límites. Las cosas de anotaciones permiten al analista agregar notas a los diagramas. Página 36. |  1. ¿Qué es un CU y cuáles son sus partes? 30 lineas  |  | | --- | | Un caso de uso describe la interacción típica entre el usuario y el sistema.  Es una técnica para capturar información de cómo un sistema trabaja actualmente, o de cómo se desea que trabaje.  Según jacobson, El CU es un documento narrativo que describe la secuencia de eventos de un actor que utiliza el sistema para completar un proceso.  **Sus partes son**: Las **precondiciones** son los hechos que se han de cumplir para que el flujo de evento se pueda llevar a cabo.  **Flujo típico de eventos**, que corresponde a la ejecución normal y exitosa del caso de uso  Los **flujos alternativos ,**son los que nos permiten indicar qué es lo que hace el sistema en los casos menos frecuentes e inesperados.  las **poscondiciones** son los hechos que se ha de cumplir si el flujo de eventos normal se ha ejecutado correctamente.  Un caso de uso provee a los desarrolladores un panorama sobre lo que desean los usuarios. Está libre de detalles técnicos o de implementación. Podemos pensar en un caso de uso como una secuencia de transacciones en un sistema.  Un caso de uso siempre describe tres cosas: un actor que inicia un evento, el evento que desencadena un caso de uso y el caso de uso que realiza las acciones desencadenadas por el evento. En un caso de uso, un actor que utiliza el sistema inicia un evento que a su vez genera una serie relacionada de interacciones en el sistema.  Los casos de uso se utilizan para documentar una transacción o evento individual. Se introduce un evento en el sistema, el cual ocurre en un tiempo y lugar específicos para provocar que el sistema haga algo.  Un caso de uso (CU) es una descripción detallada de una interacción específica entre un usuario y un sistema, que describe qué hace el sistema sin describir cómo lo hace. Un CU consta de tres partes principales: el nombre del caso de uso, la descripción del caso de uso y los actores involucrados. El nombre del caso de uso debe ser un verbo o una frase verbal que describa la acción que se realiza en el caso de uso. La descripción del caso de uso debe ser una explicación detallada de lo que sucede en el caso de uso, incluyendo los pasos que se siguen y las condiciones que deben cumplirse para que el caso de uso se complete con éxito. Los actores involucrados son las entidades externas que interactúan con el sistema y que son necesarias para que el caso de uso se complete con éxito. 36 y 6. |  1. ¿A que llamamos objeto? 25 lineas  |  | | --- | | Las personas tenemos una idea clara de lo que es un objeto: conceptos adquiridos que nos permiten sentir y razonar acerca de las cosas del mundo. Un objeto podría ser real o abstracto.  Los objetos son personas, lugares o cosas relevantes para el sistema a analizar. Los sistemas orientados a objetos describen las entidades como objetos. Algunos objetos comunes son clientes, artículos, pedidos, etcétera. Los objetos también pueden ser pantallas de GUI o áreas de texto en la pantalla  Un objeto es una instancia de una clase en un sistema orientado a objetos. Representa una entidad del mundo real o abstracto que tiene un conjunto de atributos y un conjunto de operaciones que se pueden realizar sobre esos atributos. Los objetos interactúan entre sí a través de mensajes, que son solicitudes para realizar una operación en un objeto. Cada objeto tiene un estado, que es el valor actual de sus atributos, y un comportamiento, que es el conjunto de operaciones que puede realizar. En resumen, un objeto es una entidad con un estado y un comportamiento definidos por su clase. 2 |  1. ¿Qué es un flujo de eventos en un CU? y ¿Por qué se habla de flujo típico de eventos y flujo alternativo? 30 lineas  |  | | --- | | **Flujo de eventos**: es una secuencia ordenada de operaciones para un fin determinado.  **Flujo típico de eventos**: corresponde a la ejecución normal y exitosa del caso de uso.  **Flujos alternativos**: son los que nos permiten indicar qué es lo que hace el sistema en los casos menos frecuentes e inesperados.  Un flujo de eventos en un caso de uso (CU) es una secuencia de pasos que describe la interacción entre el actor y el sistema para lograr un objetivo específico. El flujo de eventos describe lo que sucede en el sistema cuando se realiza el caso de uso, incluyendo las acciones que realiza el actor y las respuestas del sistema. El flujo de eventos también puede incluir excepciones o errores que pueden ocurrir durante la interacción. En resumen, el flujo de eventos es una descripción detallada de cómo se realiza un caso de uso en el sistema. Página 2 |  1. En un caso de uso, ¿qué información nos aporta una precondición?, y ¿una poscondición? 25 lineas  |  | | --- | | **Precondición**: describe el estado en el que debe estar el sistema para poder realizar el CU.  **Poscondición**: describe en que estado queda el sistema luego de realizarse el CU.  **Precondiciones**: la condición del sistema antes de que se pueda llevar a cabo el caso de uso, que puede ser otro caso de uso. Un ejemplo podría ser, “El espectador inició sesión con éxito en el sistema”, o podría ser la terminación exitosa de otro caso de uso.  **Postcondiciones**: o el estado del sistema después de que termine el caso de uso, incluyendo los resultados que recibieron las personas, las transmisiones a otros sistemas y los datos que se hayan creado o actualizado. Las postcondiciones se relacionan con los objetivos o requerimientos de los usuarios a partir de una definición del problema.  En un caso de uso, una precondición es una condición que debe ser verdadera antes de que se pueda iniciar el caso de uso. Es decir, es una condición que debe cumplirse para que el caso de uso tenga sentido y pueda ser ejecutado correctamente. Por ejemplo, una precondición para el caso de uso "Realizar una compra en línea" podría ser que el usuario haya iniciado sesión en el sistema y tenga una tarjeta de crédito válida asociada a su cuenta. Por otro lado, una poscondición es una condición que debe ser verdadera después de que se haya completado el caso de uso. Es decir, es una condición que debe cumplirse para que el caso de uso se considere exitoso. Por ejemplo, una poscondición para el caso de uso "Realizar una compra en línea" podría ser que se haya registrado la transacción en la base de datos del sistema y se haya enviado un correo electrónico de confirmación al usuario. En resumen, las precondiciones y poscondiciones son importantes porque ayudan a definir los requisitos y expectativas del caso de uso, y permiten verificar si se ha completado correctamente. Página 9 |  1. ¿Cuál es la utilidad de un diagrama de CU?. 15 lineas  |  | | --- | | Su utilidad radica en que nos proporciona una vista de los CU principales del sistema (funcionalidades), los actores involucrados y las relaciones entre estos y los CU.  Relación obtenida del punto 4 de la práctica.  Su utilidad radica en que describe la forma en que se utiliza el sistema.  Los diagramas de casos de uso identifican a todos los actores en el dominio del problema, de forma que un analista de sistemas se puede concentrar en lo que los humanos desean y necesitan para usar el sistema, extender sus capacidades y disfrutar su interacción con la tecnología.  Las acciones a completar también se muestran con claridad en el diagrama de caso de uso. Esto no sólo facilita al analista la tarea de identificar los procesos, sino que también ayuda en la comunicación con los demás analistas en el equipo y con los ejecutivos de la empresa.  La utilidad de un diagrama de casos de uso es proporcionar una vista general de cómo se utiliza el sistema desde la perspectiva del usuario. El diagrama de casos de uso muestra los diferentes casos de uso que el sistema puede realizar y cómo se relacionan con los actores que interactúan con el sistema. Esto ayuda a los desarrolladores a comprender los requisitos del sistema y a diseñar una solución que satisfaga las necesidades de los usuarios. Además, el diagrama de casos de uso puede servir como una herramienta de comunicación entre los desarrolladores y los usuarios, ya que proporciona una descripción visual de cómo se utilizará el sistema. En resumen, el diagrama de casos de uso es una herramienta útil para comprender los requisitos del sistema y diseñar una solución que satisfaga las necesidades de los usuarios. 7 |  1. Al redactar un CU base, si corresponde un “include” para ese CU, ¿cómo se refleja el “include” en la descripción del CU base?, puedes recurrir a un ejemplo.  40 lineas  |  | | --- | | Se refleja como una respuesta del sistema, debido a que es éste el que lo ejecuta, nunca irá como una acción de un actor determinado.  Está en un ejemplo de la práctica (gráfica).  Cuando tenemos un include en un caso de uso, en la descripción del caso de uso base mencionaremos el comportamiento que es incluido y referenciarlo, al ser un llamado al sistema está en el curso normal de los eventos. Por ejemplo los casos de uso base **registrar venta**y **ingresar producto**incluyen **buscar producto.**Cabe mencionar que los actores nunca interactúan con el caso de uso del include.  En resumen, en la descripción del caso de uso incluyente se indica la relación "include" mediante una referencia al caso de uso incluido y una descripción de cómo se utiliza en el flujo de eventos del caso de uso incluyente. Página 7. |  1. Cuál es el principal objetivo del POO? 30 lineas  |  | | --- | | Aunque la reutilización es el principal objetivo, también es muy importante el mantenimiento de los sistemas. Página 1.  En resumen, el principal objetivo de la POO es la reutilización de código y la modularidad del sistema, lo que reduce los costos de desarrollo y mejora la calidad del software. Página 1. |  1. Cómo explicamos la relación entre OBJETOS y CLASES? 35 lineas      |  | | --- | | bibliografía:  Por lo general, los objetos forman parte de un grupo de elementos similares, conocidos como clases.  Los objetos se representan y agrupan mediante clases, las cuales son óptimas para la reutilización y la facilidad de mantenimiento. Una clase define el conjunto de atributos compartidos y comportamientos que se encuentran en cada objeto de la clase. Página 2. |  1. Qué diferencia el POO del Paradigma Estructurado o Tradicional? 35 lineas  |  | | --- | | Bibliografía:  Lo que distingue a la programación orientada a objetos (y por ende al análisis y diseño orientado a objetos) de la programación clásica es la técnica de colocar todos los atributos y métodos de un objeto dentro de una estructura autocontenida, la clase en sí. Éste es un acontecimiento familiar en el mundo físico.  La herencia es una herramienta que se encuentra sólo en los sistemas orientados a objetos. |  1. Qué relación puede establecer entre los casos de uso y el análisis de sistemas? 35 lineas      |  | | --- | | Hablar de CU es hablar de análisis de sistemas, más precisamente análisis orientado a objetos.  UML se basa fundamentalmente en una técnica de análisis orientado a objetos conocida como modelado de casos de uso. Un modelo de casos de uso muestra una vista del sistema desde la perspectiva del usuario, por lo cual describe *qué* hace el sistema sin describir *cómo* lo hace. Podemos utilizar UML para analizar el modelo de casos de uso y derivar los objetos del sistema junto con sus interacciones entre sí y con los usuarios del sistema. Al utilizar técnicas de UML podemos analizar con más detalle los objetos y sus interacciones para derivar su comportamiento, atributos y relaciones. |  1. Cuáles son los principales diagramas de modelado en POO? 30 lineas      |  | | --- | | **Diagramas estructurales:**  Diagramas de clases  Diagramas de componentes  Diagramas de despliegue  **Diagramas de comportamiento:**  Diagramas de casos de uso  Diagramas de secuencia  Diagramas de comunicación  Diagramas de estados  Diagramas de actividad  Hay dos tipos principales de diagramas en UML: diagramas estructurales y diagramas de comportamiento. Los diagramas estructurales se utilizan, por ejemplo, para describir las relaciones entre las clases. Éstos se dividen en diagramas de clases, diagramas de componentes y diagramas de despliegue. Por otro lado, los diagramas de comportamiento se pueden utilizar para describir la interacción entre las personas (actores en UML) y lo que denominamos caso de uso, o la forma en que los actores utilizan el sistema. Los diagramas de comportamiento se dividen en diagramas de casos de uso, diagramas de secuencia, diagramas de comunicación, diagramas de estados y diagramas de actividad.  El diagrama de generalización/especialización (gen/ spec) es un derivado del diagrama de clases. | |