

Pregunta 1

Sin responder aún

Puntúa como 1,00

Asocie las siguientes definiciones con el tipo de prueba a la que corresponden

Se prueba la arquitectura del software y se prueban los componentes del software trabajando en conjunto

Prueba de Integración

Se prueba cada módulo del software de manera individual.

Prueba Unitaria

El software y otros elementos del Sistema como el hardware se prueban funcionando como un todo

Prueba del Sistema

Se asegura de que el software sigue los requerimientos o necesidades del cliente.

Prueba de Validación

Pregunta 2

Sin responder aún

Puntúa como 1,00

Las pruebas Alfa y Beta se suelen utilizar para:

- ☐ a. Realizar pruebas Unitarias
- ☒ b. Realizar pruebas de Validación.
- ☐ c. Realizar pruebas de Esfuerzo
- ☐ d. Realizar pruebas de Integración.

[Quitar mi elección](#)

Pregunta 3

Sin responder aún

Puntúa como 0,50

Una empresa posee un sistema legado que maneja la información de sus empleados. Entre los datos que este sistema almacena de cada empleado están su nombre completo, su cédula y la fecha en que ingresó a la empresa. Recientemente la empresa adquirió un nuevo sistema para gestionar la información de su personal. En este nuevo sistema se guarda la siguiente información de los empleados: primer y segundo nombre, primer y segundo apellido, cédula, y número de años que tiene en la empresa (entre otros). Se desea crear un mecanismo que permita que cooperen las Clases del sistema antiguo con las del sistema actual, aunque sus interfaces parezcan incompatibles. El patrón que mejor se puede aplicar en este caso es:

Seleccione una:

- ☐ a. Decorator
- ☐ b. Facade
- ☒ c. Adapter
- ☐ d. Bridge

[Quitar mi elección](#)

Pregunta 4

Sin responder aún

Puntúa como 0,50

En este patrón arquitectónico toda la información del Sistema es manejada en una base de datos central que es accesible para todos los componentes del Sistema. Los componentes no interactúan directamente, sino a través de la base de datos.

Seleccione una:

- ☐ a. Patrón MVC
- ☐ b. Arquitectura de tubos y filtros
- ☒ c. Arquitectura de repositorio
- ☐ d. Arquitectura en capas

[Quitar mi elección](#)

Pregunta 5

Sin responder aún

Puntúa como 1,00

Los desarrolladores del juego Dragon Ball FighterZ antes del lanzamiento del juego pusieron disponible para descargar un demo gratis del juego para que lo pusieran a prueba sus usuarios. Los usuarios podían reportar los errores encontrados y cualquier problema que tuvieran con la plataforma online. Este tipo de prueba más probablemente fue:

- ☐ a. Una prueba de recuperación
- ☒ b. Una prueba Beta
- ☐ c. Una prueba de regresión
- ☐ d. Una prueba Alfa

[Quitar mi elección](#)

Pregunta 6

Sin responder aún

Puntúa como 0,50

El restaurante de comidas rápidas “El Cuartel” vende dos tipos de hamburguesas: la Cuartel y la Todo Terreno. La hamburguesa Cuartel normal tiene un costo de \$7.000 y la Todo Terreno de \$10.000. Si se le adiciona queso se le suma \$1.000 y si se le adiciona cebolla grillé se le suma \$1.500. El patrón que mejor se podría aplicar para gestionar las órdenes de este restaurante es:

Tiempo restante 0:19:41

Seleccione una:

- ☐ a. Facade
- ☐ b. Proxy
- ☒ c. Decorator
- ☐ d. Adapter

[Quitar mi elección](#)

Pregunta 7

Sin responder aún

Puntúa como 0,50

Una inmobiliaria guarda información sobre sus inmuebles. La inmobiliaria maneja dos tipos de inmuebles: casas y apartamentos. La inmobiliaria puede realizar dos tipos de contrato sobre los inmuebles: venta o arriendo. Para la temporada de fin de año la inmobiliaria desea ofertar casas en venta y apartamentos para arriendo. El patrón que mejor se puede aplicar para crear estas combinaciones es:

Seleccione una:

- ☐ a. Singleton
- ☐ b. Builder
- ☐ c. Factory Method
- ☒ d. Abstract Factory

[Quitar mi elección](#)

Pregunta 8

Sin responder aún

Puntúa como 0,50

Se desea crear un programa que permita gestionar un gran número de objetos haciendo un uso eficiente de la memoria del sistema. Muchos de estos objetos tienen información idéntica o repetida. El patrón que mejor aplica en esta situación es:

Seleccione una:

- ☐ a. Composite
- ☒ b. Flyweight
- ☐ c. Adapter
- ☐ d. Facade

[Quitar mi elección](#)

Pregunta 9

Sin responder aún

Puntúa como 0,50

El lenguaje de consulta estructurada denominado SQL, es un lenguaje declarativo de acceso a bases de datos relacionales que permite especificar diversos tipos de operaciones en ellas. Existen diversos motores de bases de datos entre los cuales se pueden mencionar: Oracle, SQL Server, Sybase, DB2, etc. Estos motores utilizan distintos códigos de error para indicar fallas como: errores de clave duplicada, violación de restricciones de integridad referencial, longitud de datos, etc.

Que patrón de diseño permitiría definir un traductor de errores para cada motor de base de datos con el cual se determinaría la falla y tomarían las acciones pertinentes en función de la misma.

Seleccione una:

- ☐ a. Estrategia
- ☐ b. Builder
- ☒ c. Interpreter
- ☐ d. Abstract Factory

[Quitar mi elección](#)

Pregunta 10

Sin responder aún

Puntúa como 1,00

Las *pruebas de validación* en la ingeniería de software son el proceso de revisión que verifica que el sistema de software producido cumple con las especificaciones y que logra su objetivo, las pruebas realizadas por el usuario con el desarrollador como observador en un entorno controlado se denominan

- ☐ a. Las pruebas de aceptación
- ☐ b. Las pruebas Beta
- ☒ c. Las pruebas Alfa
- ☐ d. Las pruebas de caja negra

[Quitar mi elección](#)

Pregunta 11

Sin responder aún

Puntúa como 1,00

Es el principio básico de asignación de responsabilidades, nos indica que la responsabilidad de la creación de un objeto o la implementación de un método, debe recaer sobre la clase que conoce toda la información necesaria para hacerlo. De este modo obtendremos un diseño con mayor cohesión y así la información se mantiene encapsulada (disminución del acoplamiento). La anterior descripción hace referencia a:

- ☒ a. Patrón alta cohesión
- ☐ b. Patrón creador
- ☐ c. Patrón alto acoplamiento
- ☐ d. Patrón polimorfismo
- ☐ e. Patrón experto

[Quitar mi elección](#)

Pregunta 12

Sin responder aún

Puntúa como 1,00

La contribución más importante del GoF puede que no sea un patrón, si no una máxima que dice: «Prefiera siempre la composición de objetos antes que la herencia de clases». Entender la herencia y el polimorfismo es un reto tal, que podría empezar a otorgarle una importancia excesiva a estas técnicas. Se ven muchos diseños excesivamente complicados como resultado de ser demasiado indulgentes con la herencia - por ejemplo, muchos diseños de herencia múltiple se desarrollan por insistir en usar la herencia en todas partes. De la lectura anterior se podría concluir que al considerar un diseño de software, deberíamos cuestionarnos si:

- ☐ a. no se recomienda implementar ni herencia ni composición ya que dificulta la implementación de patrones de diseño
- ☒ b. podría ser más simple usar composición en vez de herencia, ya que ayuda a mantener cada clase encapsulada y centrada en una sola tarea.
- ☐ c. la implementación de herencia es muy importante y no podría reemplazarse en ningún caso por la composición
- ☐ d. la herencia es más fácil de implementar y más flexible

[Quitar mi elección](#)

Pregunta 13

Sin responder aún

Puntúa como 1,00

Lea con atención

Una vez generado el código el software debe ser probado para descubrir el máximo de errores posibles antes de su entrega al cliente. Es probado para descubrir errores cometidos sin darse cuenta al realizar su diseño y construcción, la pruebas requieren una mayor cantidad del esfuerzo dedicado al proyecto que cualquier otra actividad de ingeniería del software. El ingeniero crea una serie de casos de prueba que intentan "demoler" el software que ha sido construido. Por lo tanto hay que diseñar pruebas que saque a la luz diferentes clases de errores, haciéndolo con la menor cantidad de tiempo y esfuerzo. Inclusive tiene como ventaja ver hasta qué punto las funciones parecen funcionar de acuerdo con las especificaciones y cumplir así los requisitos de rendimiento.

Por tanto, existen diferentes tipos de prueba de software, entre las cuales están los tipos pruebas estructurales que se centran en los detalles procedimentales del software y a través de las cuales un ingeniero de software puede obtener casos de prueba que garanticen que se ejerciten por lo menos una vez todos los caminos independientes de cada módulo, programa o método, ejerciten todas las decisiones lógicas en las vertientes verdadera y falsa, ejecuten todos los bucles en sus límites operacionales, ejerciten las estructuras internas de datos para asegurar su validez.

Del anterior párrafo podemos afirmar que hace referencia a las pruebas

- ☐ a. Las pruebas de caja negra
- ☐ b. Las pruebas de integración
- ☐ c. Las pruebas de aceptación
- ☒ d. Las pruebas de caja blanca

Quitar mi elección

Pregunta 14

Sin responder aún

Puntúa como 1,00

Lee con atención

A la hora de definir la calidad del software se debe diferenciar entre la calidad del Producto de software y la calidad del Proceso de desarrollo. No obstante, las metas que se establezcan para la calidad del producto van a determinar las metas a establecer para la calidad del proceso de desarrollo, ya que la calidad del producto va a estar en función de la calidad del proceso de desarrollo. Sin un buen proceso de desarrollo es casi imposible obtener un buen producto.

La anterior premisa tiene sus fundamentos en

- ☐ a. Software Quality Control
- ☐ b. Total Quality Control
- ☒ c. Software Quality Assurance
- ☐ d. Total Quality Management

Quitar mi elección

Pregunta 15

Sin responder aún

Puntúa como 1,00

Para configurar un servidor en el centro de datos de la Universidad Simón Bolívar, se han definido los siguientes pasos: Instalar el sistema operativo, Instalar los parches críticos, Habilitar los servicios de red, Instalar el servidor de aplicaciones, Instalar las aplicaciones. Considerando, que de acuerdo con las necesidades del cliente, se pueden configurar las máquinas con software propietario o de código abierto, ¿Qué patrón de diseño se puede usar para ilustrar esta situación?

- ☐ a. Patrón Bridge
- ☐ b. Patrón Observer
- ☒ c. Patrón Builder
- ☐ d. Patrón Strategy

[Quitar mi elección](#)

Pregunta 16

Sin responder aún

Puntúa como 1,00

Las mejores prácticas de CMMI se enfocan en qué es necesario hacer para mejorar el desempeño y alinear las operaciones con los objetivos del negocio. Diseñado para ser comprensible, accesible, flexible e integrado con otras metodologías como las ágiles, CMMI ayuda a las organizaciones a comprender su nivel actual de capacidad y desempeño, y ofrece una guía para optimizar los resultados de negocio.

Con base en lo anterior se puede afirmar que:

- ☒ a. CMMI®, es un conjunto integrado de buenas prácticas que permiten que las empresas mejoren el rendimiento de los procesos de negocio fundamentales
- ☐ b. CMMI®, es una Metodología integral que permiten que las empresas mejoren el rendimiento de los procesos y actividades de negocio fundamentales
- ☐ c. CMMI®, es un conjunto integrado de prácticas y procesos que permiten que las empresas mejoren el rendimiento de los procesos de negocio fundamentales
- ☐ d. CMMI®, es una Metodología integrada que permiten que las empresas mejoren el rendimiento de los procesos de negocio fundamentales

Quitar mi elección

Pregunta 17

Sin responder aún

Puntúa como 1,00

CMMI es un modelo de mejora de rendimiento para organizaciones y proyectos que desean lograr un rendimiento cada vez mayor, así como abordar y resolver los desafíos de negocio. Con más de 25 años de demostrar ser globalmente efectivo en negocios y gobierno, el CMMI es un marco de trabajo integrado de mejores prácticas que pueden rápidamente mejorar y mantener el desempeño de cualquier organización para elevar su calidad, rentabilidad y competitividad.

Con base en lo anterior, sería correcto afirmar que:

- ☐ a. **CMMI es prescriptivo; describe** qué se debe hacer para mejorar las capacidades de una organización. Esto hace que el modelo sea muy flexible para cumplir las necesidades únicas de cualquier empresa.
- ☐ b. CMMI no es prescriptivo; describe qué se debe hacer para mejorar las capacidades de una organización. Esto hace que el modelo sea muy flexible para cumplir las necesidades únicas de cualquier empresa.
- ☒ c. **CMMI es prescriptivo; describe** cómo se debe hacer para mejorar las capacidades de una organización. Esto hace que el modelo sea muy flexible para cumplir las necesidades únicas de cualquier empresa.
- ☐ d. CMMI no es prescriptivo; describe cómo se debe hacer para mejorar las capacidades de una organización. Esto hace que el modelo sea muy flexible para cumplir las necesidades únicas de cualquier empresa.

Quitar mi elección

Pregunta 18

Sin responder aún

Puntúa como 0,50

Burger King es un restaurante dedicado a la preparación de comidas rápidas alrededor del mundo, su especialidad es la preparación de menús infantiles. Normalmente, estos menús están conformados por un plato principal, un acompañamiento, una bebida y un juguete. Si bien el contenido del menú puede variar, el proceso de construcción es siempre el mismo: el cajero indica a los empleados los pasos a seguir. Estos pasos son: preparar un plato principal, preparar un acompañamiento, incluir un juguete y guardarlos en una bolsa. La bebida se sirve en un vaso y queda fuera de la bolsa.

El patrón de diseño que describe la mejor solución a la anterior problemática es:

Seleccione una:

- ☐ a. Abstract Factory
- ☐ b. Singleton
- ☒ c. Builder
- ☐ d. Prototype

[Quitar mi elección](#)

Pregunta 19

Sin responder aún

Puntúa como 0,50

En programación orientada a la Web, la diferencia más importante entre el Front-End y el Back-End es que:

Seleccione una:

- ☐ a. El Front-End es estático y el Back-End dinámico
- ☒ b. El código del Front-End se ejecuta en el navegador y el código del Back-End se ejecuta en el servidor.
- ☐ c. El Front-End permite definir que elementos aparecen en una página Web y el Back-End permite modificar su apariencia

[Quitar mi elección](#)

Universidad Simón Bolívar ©2022

Pregunta 20

Institución sujeta a inspección y vigilancia por el Ministerio de Educación

Sin responder aún

Puntúa como 0,50

Este patrón de diseño combina objetos en estructuras de árbol para representar jerarquías de parte-todo. Permite que los clientes traten de manera uniforme a los objetos individuales y a los compuestos.

Seleccione una:

- ☐ a. Adapter
- ☒ b. Composite
- ☐ c. Bridge
- ☐ d. Decorator

[Quitar mi elección](#)

