



UNIVERSIDAD ABIERTA INTERAMERICANA
FACULTAD DE TECNOLOGÍA INFORMÁTICA
MATERIA: SEMINARIO DE APLICACIÓN PROFESIONAL
1er Parcial
Teórico y Práctico

Alumno: Martino Agustín Xavier

Fecha: 23/05

Tema 1

Comisión – Localización – Turno: 5-A Turno noche Sede Norte

Práctica:

Teoría:

Nota:

Temas para evaluar: Procesos y productos para la creación de software. Análisis y diseño de una aplicación web. Arquitectura de base. POO, Conocimientos de bases de datos. Tecnologías web. Patrones de diseño. Modelado.

Objetivos:

- Comprender las pautas del proceso para la creación de software desde la perspectiva de la Ing. SW.
- Comprender cómo se desarrolla una aplicación con características ingenieriles.

Modalidad: Parcial domiciliario

Requisitos para aprobar: Para que el parcial esté aprobado el alumno deberá tener correctamente desarrolladas el 60% de la teoría y resuelto el ejercicio práctico.

Tiempo:

Recomendaciones:

- a) Lea todo el parcial antes de comenzar a responder.
- b) Desarrolle una redacción clara y precisa contestando lo que la pregunta requiere.
- c) Observe la ortografía ya que la misma es parte del parcial.
- d) Si considera que no comprende alguna consigna antes de comenzar consulte a su profesor.

Notas: Las preguntas en las que se seleccionen opciones se deberá optar solo por una de las posibilidades. La indicación se efectuará con una X sobre su lateral izquierdo

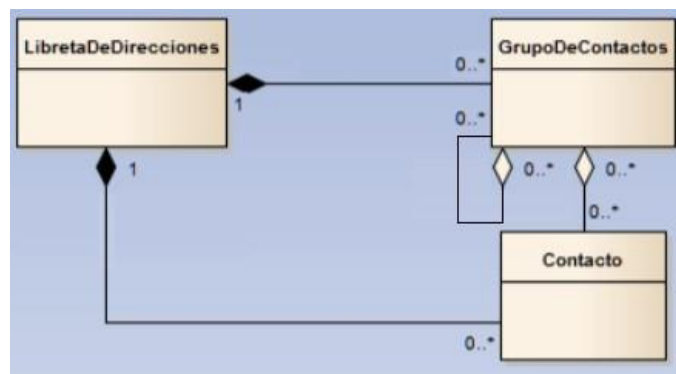
Las preguntas que solicitan justificación serán consideradas válidas si poseen la misma correctamente.

Las preguntas de múltiples posibilidades y verdadero / falso restan 0.50 puntos en caso de estar mal contestadas. En las preguntas verdadero / falso se debe tachar la opción incorrecta.

(*) la cifra entre paréntesis en cada pregunta es la cantidad de puntos sobre 100.

1. Nos solicitan desarrollar un DER a partir del siguiente diagrama de clases (Agregue los atributos que considere necesarios al modelo): (25)

(Pegar directamente la imagen DER generada con la herramienta que usted desee: por ejemplo draw.io, E.A, etc.)





UNIVERSIDAD ABIERTA INTERAMERICANA
FACULTAD DE TECNOLOGÍA INFORMÁTICA
MATERIA: SEMINARIO DE APLICACIÓN PROFESIONAL
1er Parcial
Teórico y Práctico

2. ¿Cómo podría relacionar la plataforma Auth0 (Similitudes/diferencias) con el concepto del patrón (Composite) Usuario-Patente-Familia? (20)

La plataforma Auth0 y el patrón composite no tienen una relación directa entre sí, ambos tienen dos enfoques distintos: Por un lado el auth0 está enfocado en la gestión de identidades y accesos que se proporcionan para las soluciones que requieran autenticación y autorización de usuarios sobre las aplicaciones y servicios soportados por sitios webs. En cambio el patrón composite está enfocado en crear estructuras jerárquicas sobre ciertos objetos ejemplo un usuario que contiene grupos de permisos que contienen permisos, los cuales no requieren tener la relación dentro de la base de datos, sino con tener una relación con el grupo, ya obtiene los permisos que el grupo contenga. Por más que no contengan una relación directa se pueden implementar mutuamente donde el manejo de permisos y roles de usuario esté manejado por un patrón composite y el acceso y autenticación esté manejado mediante la plataforma de Auth0

3. Enumere los elementos que se utilizan en un caso de uso y qué representan. (5)

Los elementos que se utilizan en un caso de uso son:

El stakeholder: representa quien es el usuario o persona que utilizará la funcionalidad

El caso de uso: describe la funcionalidad específica del sistema

Escenario: Describe la secuencia específica de acciones que se realizarán sobre el caso de uso

Diagrama de casos de uso: representa todas las funcionalidades del sistema con casos de uso y todos los actores o stakeholders que se relacionan con el sistema

Requisitos funcionales: son aquellos requerimientos los cuales son especificaciones detalladas de las características del sistema

Requisitos no funcionales: Son aquellas especificaciones detalladas que no están relacionadas directamente con las funcionalidades

4. ¿Qué son los modelos prescriptivos de software? (5)

Los modelos prescriptivos de software son un enfoque basado en la definición en conjunto de los procesos y actividades que deben utilizarse para realizar software de alta calidad. Estos modelos mayormente están centrados en la planificación, control y documentación sobre el proceso de desarrollo de software

5. ¿UP a qué proceso prescriptivo pertenece? (5)

UP es un proceso de desarrollo de software iterativo e incremental, el cual tal como comentamos corresponde al modelo iterativo el cual el proceso se divide en ciclos, donde se va haciendo enfoque sobre una funcionalidad en específica dentro de cada uno de estos ciclos o iteraciones

6. ¿Cuáles son las disciplinas de UP? (5)

Las disciplinas son 7:

- Modelado de negocio
- Requerimientos
- Análisis y diseño del sistema
- Implementación del sistema
- Pruebas
- Despliegues
- Gestión de proyecto

7. ¿A qué denominamos arquitectura del software? (10)

Denominamos arquitectura de software al diseño de cómo vamos a implementar la solución, es decir, de qué forma las distintas funcionalidades se van a comunicar, cómo van a enviarse los mensajes, de qué forma se va a poder acceder, y lo más importante hasta qué punto podemos llegar a escalar el tráfico y uso de nuestro software ya que sin una buena arquitectura cualquier mensaje o string de más puede llegar a realizar un congestionamiento el cual puede generar problema en la solución

8. ¿Qué es el producto de software? (5)

El producto de software es aquello que tomamos como resultado final del proceso del desarrollo de software



UNIVERSIDAD ABIERTA INTERAMERICANA
FACULTAD DE TECNOLOGÍA INFORMÁTICA
MATERIA: SEMINARIO DE APLICACIÓN PROFESIONAL
1er Parcial
Teórico y Práctico

9. ¿Cuáles son las actividades estructurales que propone el proceso de software? (10)

Las actividades que propone el proceso de software son:

Especificación de requerimientos, Análisis sobre el problema o necesidad, Diseño de la solución, Implementación de la solución, Pruebas, Mantenimiento, Gestión del proyecto (monitoreo, seguimiento y control), Documentación: implica tener el software documentado donde incluye los manuales tanto de usuario como del programador

10. ¿En qué se diferencian un Hashtable y un Dictionary en C#? (10)

En lo que se diferencian son en:

Tipo de clave: Dictionary solo admite tipos de clave que sean objetos mientras que hashtable admite cualquier tipo de clave

Rendimiento: Hashtable se basa principalmente en una tabla hash mientras que por otro lado Dictionary utiliza una estructura de árbol para almacenar los pares de clave valor

Capacidad de sincronización: Hashtable es una clase thread-safe y se puede sincronizar fácilmente, en cambio, Dictionary no está diseñado para entornos multi-hilo

PRÁCTICA

- 1) Desarrollar un diagrama de clases (Sólo del dominio) y el programa donde se observe claramente los campos, propiedades y métodos para un sistema de logística donde nos informan que existen varios depósitos y tiendas donde sus principales datos son: Número de identificación, nombre, provincia y localidad. Las acciones que se pueden desarrollar son: Transferencia de stock de diversos productos entre depósitos y transferencia de stock desde depósitos a tiendas. Los depósitos y tiendas cuentan con ciertos productos de los cuales conocemos su nombre, descripción y precio. Cada histórico de operación deberá contener: Origen, destino, fecha, usuario y el listado de productos con sus cantidades transferidos. Se deberá contar con al menos un usuario registrado en el sistema para realizar las operaciones
- 2) Generar un modelo de Repositorio accesible a través de una WebApi (Con los métodos necesarios para las operaciones solicitadas) para persistir la información de transferencias entre los diversos depósitos/tiendas y luego poder consultar las mismas. Puede utilizar cualquier modelo de persistencia, incluso en memoria, para las entidades intervinientes.
- 3) El modelo de solución generado deberá contener mínimamente 5 capas (Sitio web para al cliente, WebApi, Business Layer, DomainModel y Repositorio) asignando las responsabilidades correspondientes. Utilizar los patrones de diseño y arquitectura que usted crea conveniente.
- 4) Deberá generar casos de prueba para testear las operaciones solicitadas. Mínimamente se podrá:
 - a. Se tienen que poder filtrar los tipos de operaciones a partir de un destino (Elegir un depósito o tienda) y luego mostrar: Origen, producto, cantidad, usuario y fecha, esta última ordenada de manera descendente.
 - b. Se debe contar con otro informe que nos permita filtrar los destinos que solicitaron más de tres operaciones en el día, mostrando origen, destino, producto, cantidad y usuario que los solicitó ordenado por fecha de manera ascendente.
 - c. A partir de un nro de tienda, mostrar los primeros 10 movimientos entre las fechas determinadas por el usuario (Elija los datos que desea mostrar para esta consulta).
- 5) Entre la WebApi y el sitio web se dialogará con formato de mensajes en json. Puede utilizar cualquier arquitectura en la web (Razor, Blazor, HTML5 puro con invocaciones Ajax, etc.). También será valorado el diseño CSS y javascript que permita una buena usabilidad.

Firma del Alumno