**Alumno**: Pablo Daniel Mussanti **Fecha: 26/5/2020 Tema 1**

**Comisión – Localización - Turno:5ª Sede Norte T. Noche**

**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:**

1. Lea todo el parcial antes de comenzar a responder.
2. Desarrolle una redacción clara y precisa contestando lo que la pregunta requiere.
3. Observe la ortografía ya que la misma es parte del parcial.
4. 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

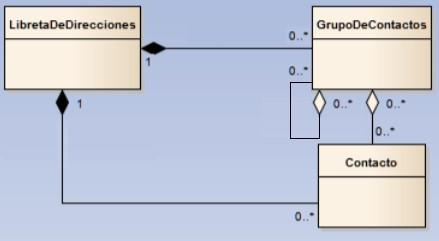
Las preguntas que solicitan justificación serán consideradas válidas si poseen la misma correctamente.

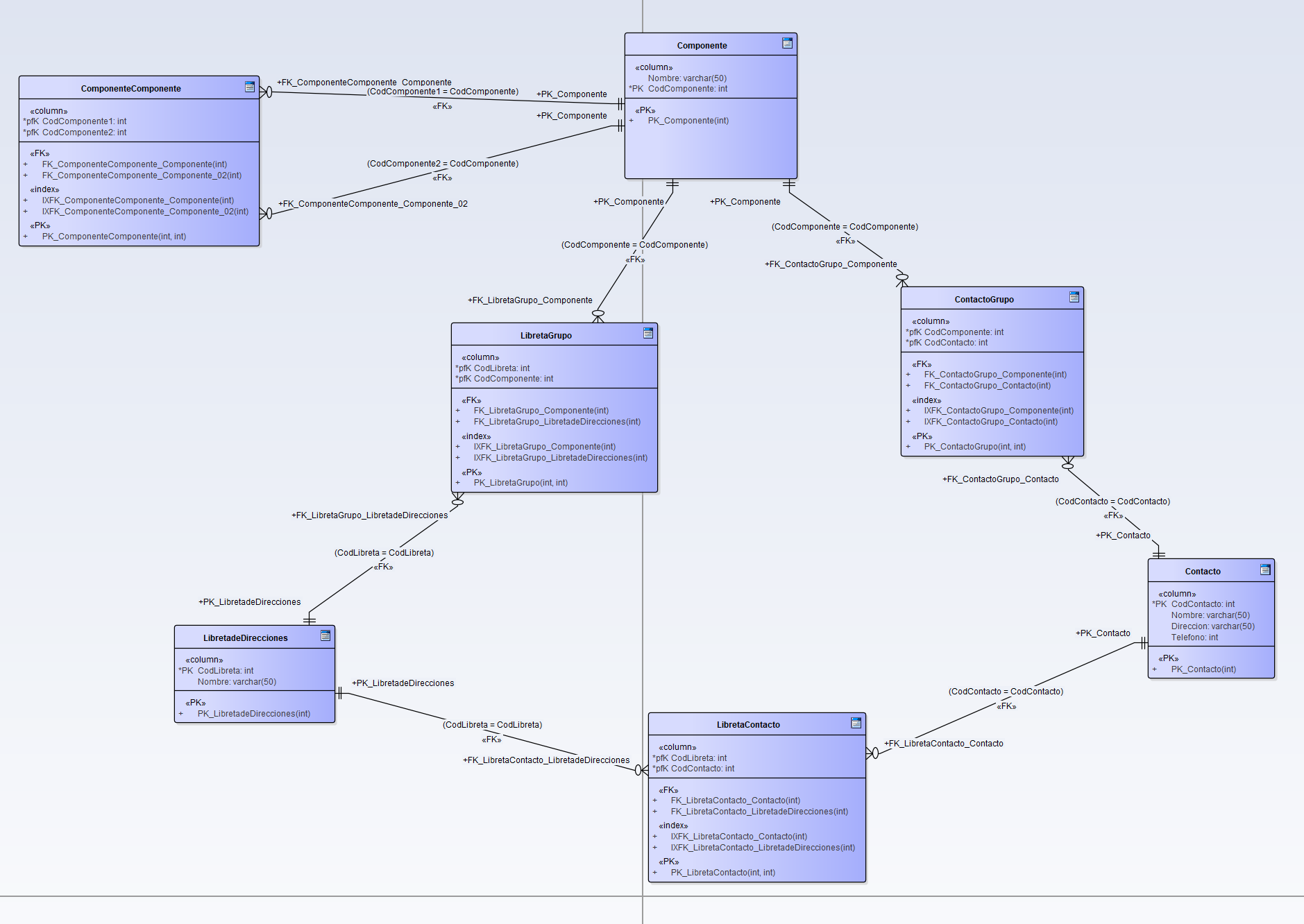
Las preguntas de múltiples posibilidades y verdadero / falso restan 0.50 puntos en caso de estar mal contestadas. En las preguntas verdadero / falso se debe tachar la opción incorrecta.



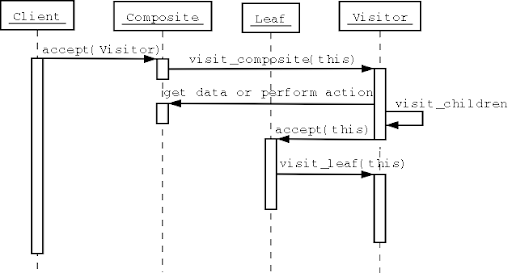
(\*) la cifra entre paréntesis en cada pregunta es la cantidad de puntos sobre 100.

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.)

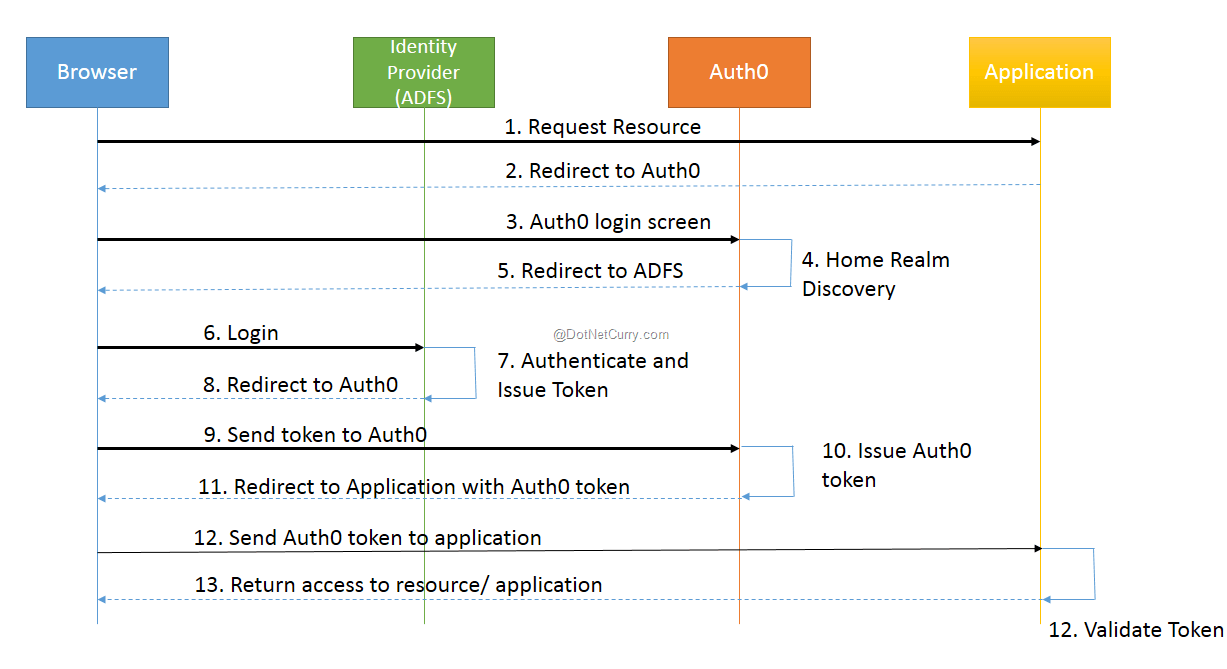




1. ¿Cómo podría relacionar la plataforma Auth0 (Similitudes/diferencias) con el concepto del patrón (Composite) Usuario-Patente-Familia? (20)

Composite  


Auth0



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

Se pueden determinar:

* **Sistema:** Rectángulo que representa los límites del sistema, este contiene a los casos de uso.
* **Actor:** Persona/Ente que interactúa con el sistema
* **Caso de uso:** Operación o tarea que se realiza específica que se realiza tras una orden de algún agente externo
* **Relaciones:** Las relaciones que se dan entre un actor y un caso de uso o un caso de uso con otro.

Estos pueden ser:

* **Actor - CU:**
* **Comunicación**
* **CU - CU:**
* **Include** (el CU contiene un comportamiento común para otro CU)
* **Extend** (el CU maneja las excepciones del CU básico)
* **Use** (conjunto de características que son similares en más de un CU y no se desea mantener copiada la descripción de esta característica)
* **Dependencia** (un CU depende de otro)

1. ¿Qué son los modelos prescriptivos de sw? (5)

Los modelos prescriptivos de SW se lo definen como modelos que comprenden un conjunto de actividades, acciones tareas, fundamentos y productos de trabajo que se requieren para desarrollar software de alta calidad. Estos proporcionan estabilidad, control y organización.

El marco de trabajo usual es:

* Comunicación.
* Planeación.
* Modelado.
* Construcción.
* Desarrollo.

Los podemos clasificar en:

* Lineales.
* Incrementales.
* Evolutivos.

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

El Proceso Unificado es Iterativo e Incremental

Desarrollar un producto de software comercial es una tarea enorme que puede continuar por varios meses o años. Es práctico dividir el trabajo en pequeños pedazos o mini-proyectos. Cada mini-proyecto es una iteración que finaliza en un incremento. Las iteraciones se refieren a pasos en el flujo de trabajo, los incrementos se refieren a crecimiento en el producto. Para ser más efectivo, las iteraciones deben estar controladas, esto es, deben ser seleccionadas y llevadas a cabo de una manera planeada.

Los desarrolladores basan su selección de qué van a implementar en una iteración en dos factores. Primero, la iteración trata con un grupo de casos de uso que en conjunto extienden la usabilidad del producto. Segundo, la iteración trata con los riesgos más importantes. Las iteraciones sucesivas construyen los artefactos del desarrollo a partir del estado en el que fueron dejados en la iteración anterior.

En cada iteración, los desarrolladores identifican y especifican los casos de uso relevantes, crean el diseño usando la arquitectura como guía, implementan el diseño en componentes y verifican que los componentes satisfacen los casos de uso. Si una iteración cumple sus metas – y usualmente lo hace – el desarrollo continúa con la siguiente iteración. Cuando la iteración no cumple con sus metas, los desarrolladores deben revisar sus decisiones previas y probar un nuevo enfoque.

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

Se destacan:

* Inicio - análisis.
* Elaboración - diseño.
* Construcción - implementación.
* Transición - pruebas.

Características del UP:

* Centrado en su arquitectura.
* Enfoque en riesgos.
* Iterativo e incremental.
* Dirigido por casos de uso.

1. ¿A qué denominamos arquitectura del software? (5)

La [arquitectura](https://www.ecured.cu/Arquitectura) de [software](https://www.ecured.cu/Software) es un conjunto de patrones que proporcionan un marco de referencia necesario para guiar la construcción de un software, permitiendo a los [programadores](https://www.ecured.cu/index.php?title=Programadores&action=edit&redlink=1), [analistas](https://www.ecured.cu/index.php?title=Analistas&action=edit&redlink=1) y todo el conjunto de desarrolladores del [software](https://www.ecured.cu/Software) compartir una misma línea de trabajo y cubrir todos los objetivos y restricciones de la aplicación.

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

Según la IEE “software es la suma total de los programas de ordenador, procedimientos, reglas, la documentación asociada y los datos que pertenecen a un sistema de cómputo" y "un producto de software es un producto diseñado para un usuario”

1. Enumere y defina para qué sirve cada uno de los diagramas de comportamiento que posee UML. (10)

Los principales diagramas definidos dentro de UML son:

* **Diagrama de clases:** Define la estructura de un sistema, sus clases y las relaciones entre estas.
* **Diagrama de secuencia:** Muestra el comportamiento y las interacciones de los objetos durante el tiempo.
* **Diagrama de caso de uso:** Modelan las funcionalidades del sistema desde una perspectiva externa.
* **Diagrama de actividad:** Muestra el flujo de actividades que realiza el sistema.
* **Diagrama de estado:** Representa comportamientos y situaciones durante la vida de un objeto.

1. ¿Cuáles son las actividades estructurales que propone el proceso de software? (5)

La Estructura del Proceso consta de 5 actividades:

1. **COMUNICACIÓN:** Antes de que comience cualquier trabajo técnico tiene importancia crítica comunicarse y colaborar con el cliente (y con otros participantes). Se busca entender los objetivos de los participantes respecto del proyecto y reunir los requerimientos que ayuden a definir las características y funciones del software.
2. **PLANEACIÓN:** La actividad de planeación crea un mapa que guía al equipo, este mapa es llamado plan del proyecto del software, define el trabajo de ingeniería de software al describir las tareas técnicas para realizar los riesgos probables, los recursos que se requieren, los productos del trabajo que se obtendrán y una programación de las actividades.
3. **MODELADO:** Un diseñador, arquitecto, carpintero, etc. crea un bosquejo del objeto por hacer a fin de entender el panorama general, cómo se verá arquitectónicamente, cómo ajustan entre si las partes constituyentes y muchas características más, un ingeniero de software hace lo mismo al crear modelos a fin de entender mejor los requerimientos del software y el diseño que los satisfará.
4. **CONSTRUCCIÓN:** Esta actividad combina la generación de código (ya sea manual o automatizada) y las pruebas que se requieren para descubrir errores en éste.
5. **DESPLIEGUE:** El software se entrega al consumidor que lo evalúa y le da retroalimentación, misma que se basa en dicha evaluación.
6. ¿Qué características posee el desarrollo Ágil? (5)

Las metodologías ágiles son aquellas que permiten adaptar la forma de trabajo a las condiciones del proyecto, consiguiendo flexibilidad e inmediatez en la respuesta para amoldar el proyecto y su desarrollo a las circunstancias específicas del entorno.

Los MA son una familia de técnicas de desarrollo diseñadas para entregar productos a tiempo, con presupuesto, y con alta calidad y satisfacción del cliente.

1. ¿En qué se diferencian un Hashtable y un Dictionary en C#? (5)

Podemos determinar:

Una diferenciación en su definición, almacenamiento, requerimientos, la recuperación de datos, resultados, organización y rendimiento.

Dictionary:

* Colección genérica
* Se define en System.Collections.Generic
* Puede almacenar pares clave/valor del mismo tipo
* Se debe especificar el tipo de clave y valor
* La recuperación de datos es mas rapida
* Si uno intenta acceder a una clave que no está presente en el diccionario, dará un error
* Menos rendimientos en hilos
* Mantiene el orden de los valores almacenados

Hashtable:

* Colección no genérica
* Se define en System.Collections
* Puede almacenar pares clave/valor del mismo tipo o de un tipo diferente
* No es necesario especificar el tipo de clave y valor
* Recuperación de datos lenta
* Si uno intenta acceder a una clave que no está presente en el hashtable, dará null
* Mejor rendimiento en múltiples hilos
* No mantiene el orden de los valores almacenados

**PRÁCTICA**

* 1. Desarrollar un diagrama de clases (Solo el Domain Model) y una solución web donde se observen claramente los campos, propiedades y métodos para un sistema de cuentas bancarias donde nos informan que existen dos tipos de cuentas: Las cajas de ahorro CA y las cuentas corrientes CC. Ambas poseen número de cuenta y saldo. Las acciones que se pueden desarrollar son Depósitos, Extracciones y Transferencias. Una diferencia importante entre las CA y las CC es que las CC permiten un descubierto.
  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 depósitos, extracciones y transferencias entre cuentas 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 4 capas (Sitio web para al cliente, WebApi, 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: Consultar el saldo de una cuenta, depositar y extraer dinero, transferir hacia otra cuenta y consultar los últimos movimientos de una cuenta. Si se desea agregar una capa de autenticación será valorado.

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

