**Examen Programador Java**

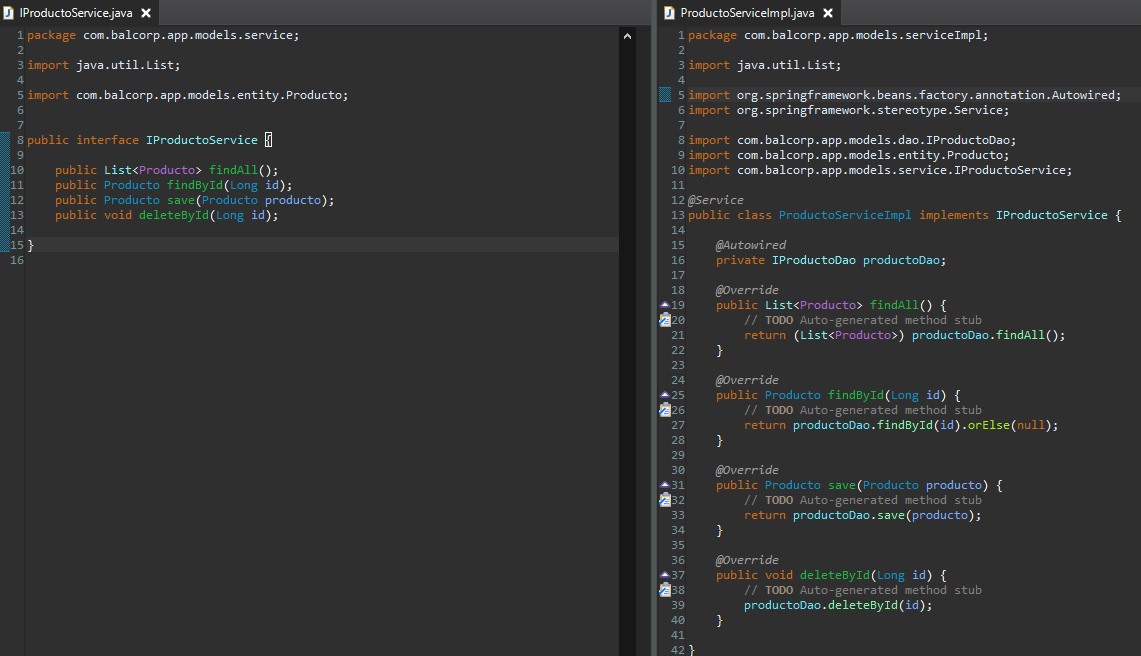
**Nombre: Christian Baldeón Baldeón**

**Fecha: 29/11/2020**

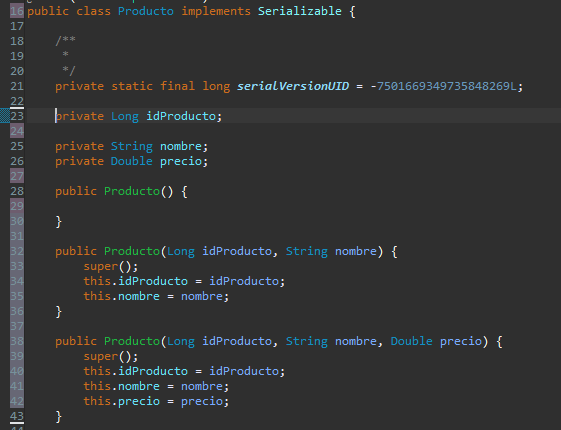
Preguntas:

1. Grafique y codifique un ejemplo de: (3 pts)
   1. Herencia:
   2. Interface:
   3. Polimorfismo
2. ¿Qué es un método overriding and overloading, codifique un ejemplo de cada uno? (2 pts)

**Método overriding** métodos que sobre escriben métodos de la clase base (mismo nombre, misma cantidad y tipo de parámetros) en cambio un método **overloading** puede tener mismo nombre, pero diferente cantidad de parámetros o diferente tipo de parámetros en la clase hija con respecto a la clase base.

 **Ejemplo overriding**

Ejemplo **overloading por constructor**

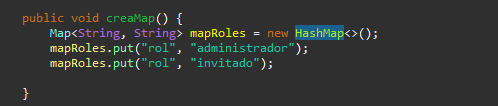


1. ¿Cuál es la diferencia entre una interface y una clase abstracta? (1 pts)

La interface tiene la capacidad de poder ser implementada junto a otros interfaces en clases hijas en cambio en las clases abstractas solo se permite extender una en la clase hija

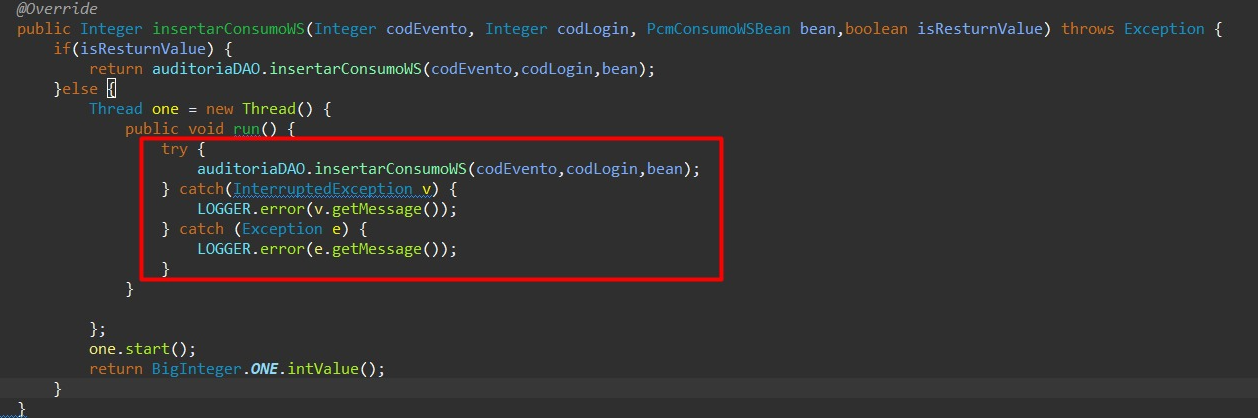
1. ¿Cuáles son las clases e interfaces que están disponibles en las colecciones? (1 pts)
2. ¿Explique cómo se usa el Map, codifique un ejemplo? (2 pts)

Interface para almacenar elementos clave/valor



1. ¿Qué es la propagación de Excepciones?, codifique un ejemplo (2 pts)

Mecanismo de captura de errores con la finalidad de asegurar la continuidad de la ejecución de las instrucciones siguientes a las líneas de código con la posible excepción, también se procura el manejo amigable de excepciones para el cliente final además de salvaguardar la información de las tecnologías usadas ante posibles atacantes.



1. Describa la semántica de versionamiento aplicada a microservicios (2 ptos)

Estandar de versionamiento de APIs basada en magnitud de cambios, por ejemplo

2.3.5

Un cambio en la posición del número 2 representa cambios drásticos en la API con posibles incompatibilidades con versiones anteriores

Un cambio en la posición del número 3 representa mejoras compatibles con versiones anteriores

Un cambio en la posición del número 5 representa arreglos realizados a errores en la versión anteriores

1. Qué entiende por Frontend y Backend (1 pto)

Frontend: Aplicativos orientados a la interacción con el usuario final

Backend: Aplicativos que contienen la lógica del funcionamiento del sistema

1. Describa cuales son los estados del ciclo de vida de un hilo (2 pts)

Nuevo hilo: hilo instanciado sin recursos asignados

Ejecutable: estado del hilo luego de llamar al método start, en este estado el hilo ya cuenta con recursos para empezar su ejecución.

Parado: Estado en el que el hilo detiene su ejecución en consecuencia de llamada a los métodos sleep, wait y otros.

Muerto: muere el hilo cuando termina la ejecución de su método run o cuando se invoca al método stop.

1. ¿Cuál es la diferencia entre la clase Thread y la interface Runnable para crear un hilo? (2 pts)
2. ¿Cuáles son los niveles de log que se manejan en Java y qué componente utiliza?, codifique un ejemplo (2 pts)

En log4j2 se manejan los niveles:

All, debug, info, warn, error, fatal y off

1. ¿Qué es synchronization, codifique un ejemplo (2 pts)
2. ¿Cuáles son los frameworks en Java para implementar SOAP Web Services? (1 pts)

Apache CXF

Spring Web Services

1. Defina el stack del protocolo web services y sus capas (2 pts)
2. ¿Cuáles son los elementos de un mensaje SOAP? (1 pts)
3. ¿Qué es el spring security?, describa un ejemplo (1 pts)

Framework para el control de acceso, proporciona auntenticación y autorización para aplicaciones java

1. ¿Cuáles son los tipos de respuesta en un WSDL? (1 pts)
2. ¿Qué es el JAX-WS? (1 pts)

Librería para crear y consumir servicios soap

1. Codifique un elemento SOAP envelope (2 pts)
2. ¿Qué es el WADL? (1 pts)

XML que representa el contrato de un servicio REST similar a un contrato WSDL en servicios soap

1. Dada las siguientes tablas (2 pts):

*CREATE TABLE dbo.envelope(id int, user\_id int);*

*CREATE TABLE dbo.docs(idnum int, pageseq int, doctext varchar(100));*

*INSERT INTO dbo.envelope VALUES*

*(1,1),*

*(2,2),*

*(3,3);*

*INSERT INTO dbo.docs(idnum,pageseq) VALUES*

*(1,5),*

*(2,6),*

*(null,0);*

¿Cuál es la salida del siguiente query?

*UPDATE docs SET doctext=pageseq FROM docs INNER JOIN envelope ON envelope.id=docs.idnum*

*WHERE EXISTS (*

*SELECT 1 FROM dbo.docs*

*WHERE id=envelope.id);*

El query actualiza el campo doctext con el valor de pageseq de la table docs para los idnum igual a 1 y 2, la tabla actualizada docs queda de la siguiente forma

|  |  |  |
| --- | --- | --- |
| docs | | |
| Idnum | Pageseq | doctext |
| 1 | 5 | 5 |
| 2 | 6 | 6 |
| Null | 0 | null |

1. Dada las siguientes tablas (2 pts):

*SELECT \* FROM users;*

*user\_id username*

*1 John Doe*

*2 Jane Don*

*3 Alice Jones*

*4 Lisa Romero*

*SELECT \* FROM training\_details;*

*user\_training\_id user\_id training\_id training\_date*

*1 1 1 "2015-08-02"*

*2 2 1 "2015-08-03"*

*3 3 2 "2015-08-02"*

*4 4 2 "2015-08-04"*

*5 2 2 "2015-08-03"*

*6 1 1 "2015-08-02"*

*7 3 2 "2015-08-04"*

*8 4 3 "2015-08-03"*

*9 1 4 "2015-08-03"*

*10 3 1 "2015-08-02"*

*11 4 2 "2015-08-04"*

*12 3 2 "2015-08-02"*

*13 1 1 "2015-08-02"*

*14 4 3 "2015-08-03"*

Escribe un query que muestre los usuarios que tomaron una lección de entrenamiento más de una vez en el mismo día agrupado por usuario y lección de entrenamiento, ordenada desde la lección más reciente a la más antigua.

select t.user\_id, u.user\_name, t.training\_id, t.training\_date, count(training\_id) as number\_repeted\_lessons

FROM dbo.training\_details t inner join dbo.users u on t.user\_id = u.user\_id

GROUP BY t.user\_id, t.training\_id, t.training\_date,u.user\_name

HAVING COUNT(t.training\_date)>1

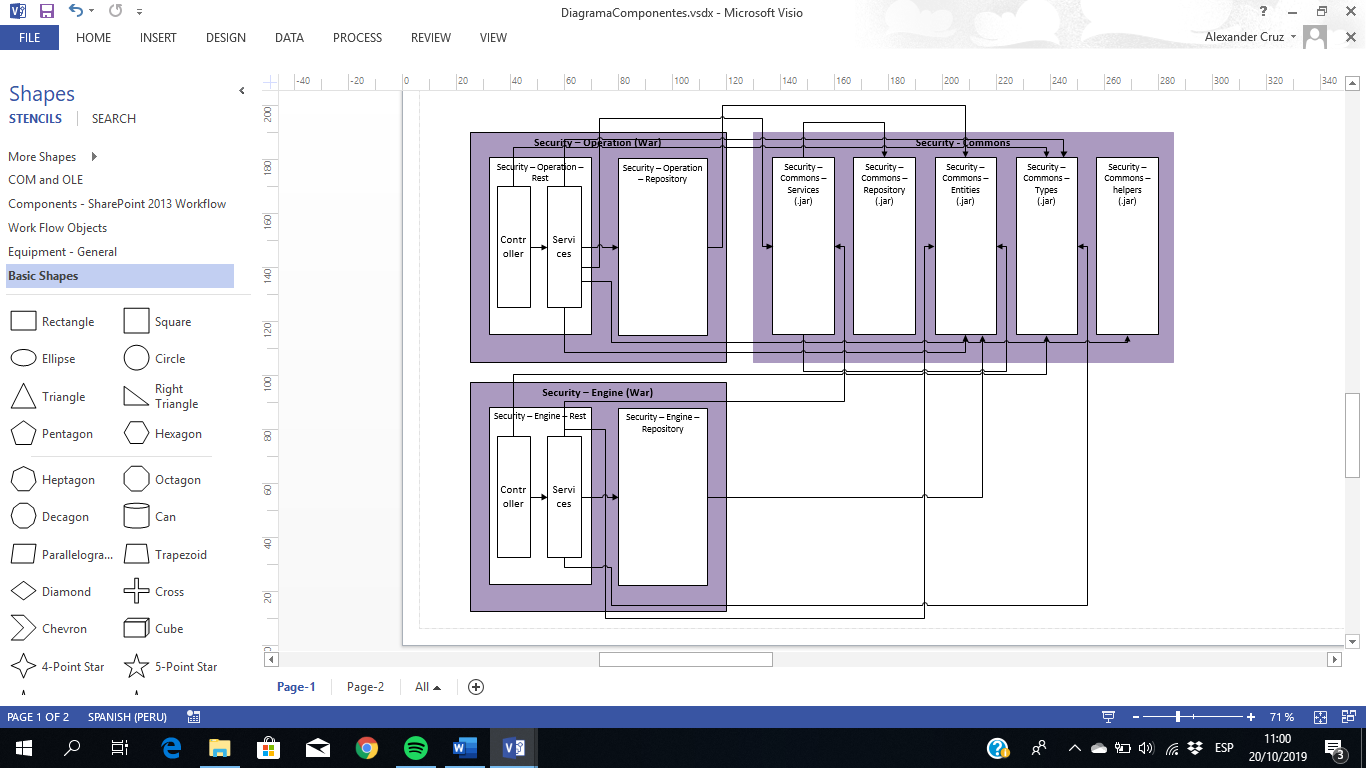
order by t.training\_date desc

1. Se le solicita implementar dos servicios: (24 ptos)
   1. Que permite hacer un CRUD de usuarios
   2. Que permite autenticar y autorizar al usuario

Defina el modelo de datos y codifique, usando Java, los servicios utilizando todas las capas que considera necesarias; para ello el arquitecto de la empresa le entrega una arquitectura estándar de capas, la cual se recomienda utilizar en su desarrollo.

Nota:

* Los nombres de cada capa de desarrollo son bastantes sugerentes
* En caso no tenga muy claro la arquitectura presentada, utilice sus propias capas y fundamente que hace cada una.
* Puede utilizar SOAP o REST
* Se tomará en cuenta el manejo de excepciones, log, mensajes, códigos de error, optimización y limpieza del código



La implementación de los servicios solicitados se encuentra en la carpeta servicios