

Nombre: Maximiliano Cazares Martínez

1. Que es la inyección de dependencia y representa los tipos de inyección de dependencia que existen.

Es el principio de poder pasar las dependencias cuando sea necesario en lugar de inicializar las dependencias en la clase receptora.

Inyección con setters

```
Class claseA{
    claseB B;

    void setClaseB(claseB inyectada){
        B = inyectada;
    }

    int diezPorciento(){
        return B.calcular() * 0.10;
    }
}

class main {
    Public static void main(){
        claseA A = new claseA();
        extraClaseB extraB = new extraClaseB();
        claseA.setClaseB(extraClaseB);
    }
}
```

Inyección con constructor

```
Class claseA{
    claseB B;

    claseA(claseB inyectada){
        B = inyectada;
    }

    int diezPorciento(){
        return B.calcular() * 0.10;
    }
}

class main {
    Public static void main(){
        claseB B = new claseB();
    }
}
```

```

        claseA A = new claseA(B);
    }
}

```

2. Menciona al menos 5 patrones de diseño y para que los utilizas.

Singleton: Garantiza la existencia de una única instancia para una clase

Builder: Ayuda a crear objetos complejos de manera sencilla y escalable.

Bridge: Separar la abstracción de la implementación

Facade: Nos provee una interfaz unificada y simple para un sistema complejo.

Observer: Un objeto le pasa el estado interno a muchos objetos que están interesados.

3. Representa y explica cada uno de los componentes de la arquitectura del framework Spring.

4. Describe para que se utilizan las siguientes anotaciones

@Autowired: Marca una dependencia que Spring resolverá e inyectará.

@Qualifier: Se utiliza junto con @Autowired para proporcionar el bean id o el bean name que queremos usar en situaciones ambiguas.

@PostConstruct:

@PreDestroy:

@Required: Se utiliza en los setters para marcar dependencias que queremos poblar con XML

@Bean: Marca un factory method que instancia un Spring Bean

@Configuration: Establece que existen uno o más beans con los que trabajar en tiempo de ejecución.

@Transactional: Permite realizar una transacción sobre una base de datos

@Secured: Especifica que un método debe ser ejecutado solo si el usuario autenticado tiene la autoridad necesaria

@RolesAllowed: Especifica la lista de roles permitidos para acceder a los métodos en una aplicación

@ManagedResource: Indica al registro de instancias de una clase con servidores JMX.

@ManagedAttribute: Indica la propiedad bean para exponer como un atributo JMX

@ManagedOperation: indica un método dado como una operación JMX

@Cacheable: Indica que el resultado de invocar un método puede ser atrapado

5. Para que sirve un converter.

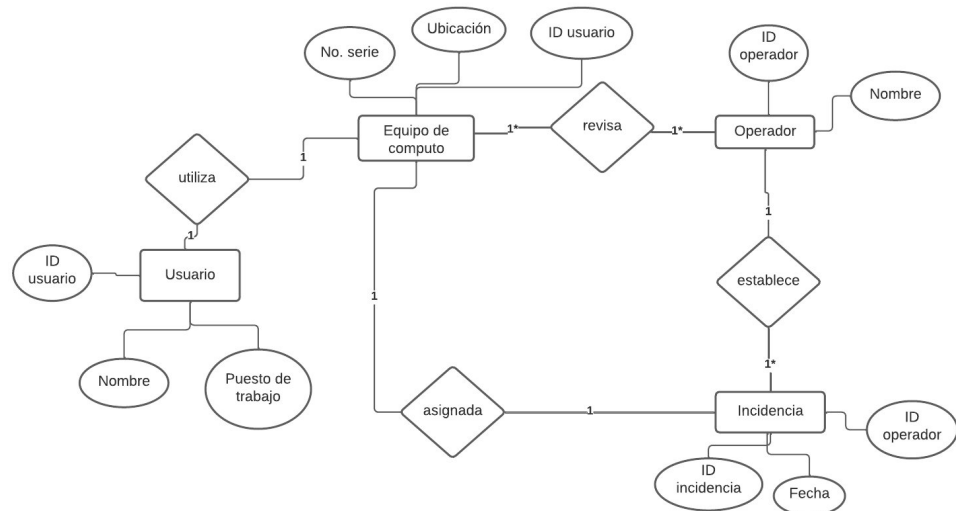
Es una interfaz que describe que una clase puede ejecutar conversiones objeto-string y string-objeto.

6. Que es un aspecto y describe un ejemplo de cómo lo podríamos implementar

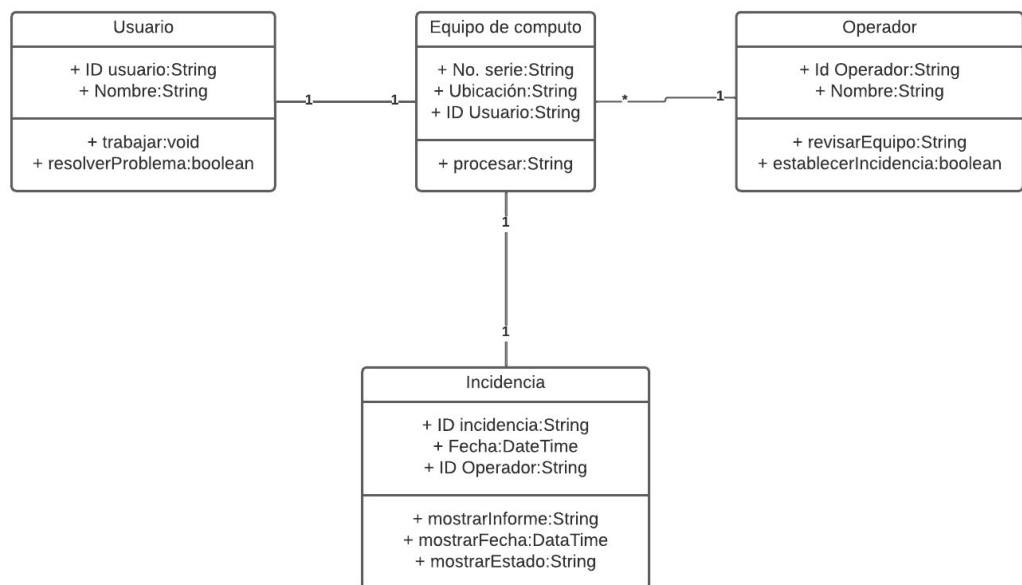
7. La empresa XYZ es la encargada de administrar el reporte de incidencias de los equipos de cómputo, donde un operador puede dar de alta una incidencia ("definir los datos que

consideres más importante ”), consultar incidencias por ID, por día , y por operador que las dio de alta , cabe mencionar que debemos tener dashboards que representen número de peticiones a los servicios consultados , insertados , cuantos se resolvieron , cuantos están abiertos.

Realizar el diagrama E-R



Realizar el diagrama de clases y programar un servicios rest de consulta e inserción que consideren mas importantes.



8. Que es la arquitectura de micro servicios.

Es el enfoque de desarrollo de una aplicación de servidor como un conjunto de servicios pequeños. Donde cada servicio se ejecuta en su propio proceso y se comunica con otros procesos mediante protocolos como HTTP/HTTPS. Esta principalmente orientado al back-end pero también ser utilizado en el front-end.

9. Menciona para que sirve cada uno de los siguientes componentes

Spring Cloud Config: Es un servicio que provee apoyo tanto para cliente como para servidor para la configuración externa en un sistema distribuido.

Eureka:

Ribbon:

Hystri:

Zuul:

OAUT:

SWAGGER:

SPLUNK:

10. Generar una tarea ya sea con MAVEN o GRADLE para generar las clases de un WSDL del mismo.