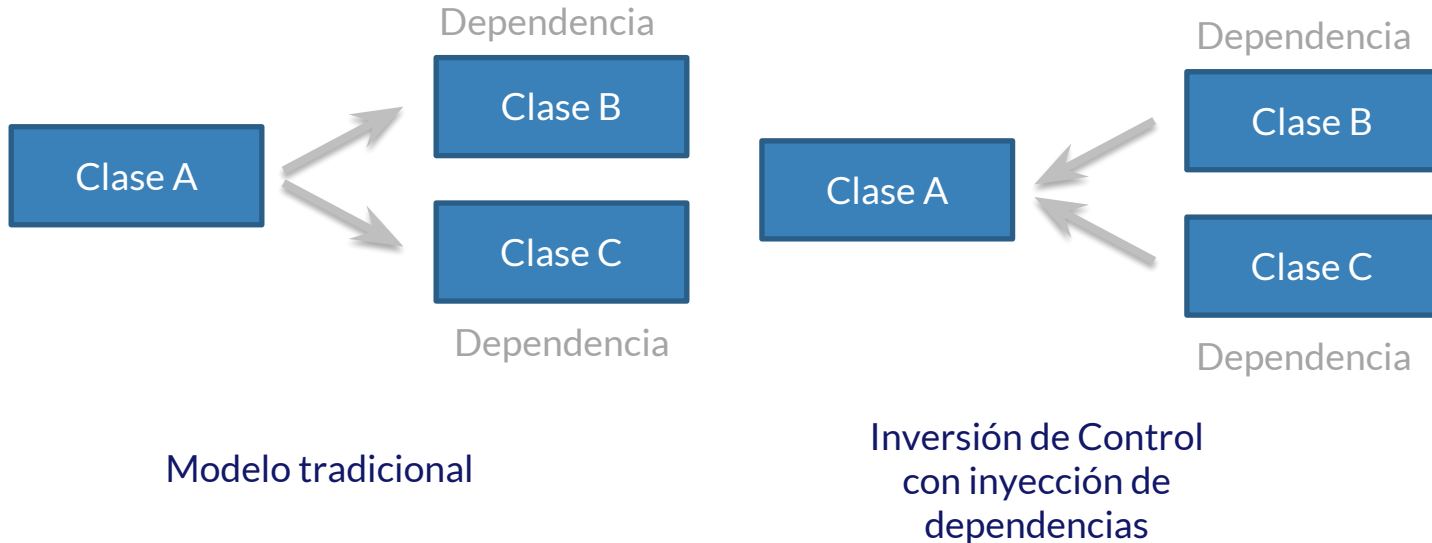


# INYECCIÓN DE DEPENDENCIAS



# INYECCIÓN DE DEPENDENCIAS

- ▶ Es una forma de inversión de control.



# TIPOS DE INYECCIÓN DE DEPENDENCIAS

- ▶ Vía *setter* (o *property*)
- ▶ Vía constructor
- ▶ Referencias entre beans

# INYECCIÓN VÍA SETTER

```
public class Saludator {  
    private String mensaje;  
    public void setMensaje(String str) {  
        this.mensaje = str;  
    }  
    //...  
}
```

```
<beans xmlns="http://www.springframework.org/schema/beans"... >  
    <bean id="saludator"  
        class="com.openwebinars.beans.Saludator">  
        <property name="mensaje"  
            value="Hola alumnos de openwebinars"> </property>  
        </bean>  
</beans>
```

# INYECCIÓN VÍA CONSTRUCTOR

```
public class Saludator {  
    private String mensaje;  
    public Saludator(String str) {  
        this.mensaje = str;  
    }  
    //...  
}
```

```
<beans xmlns="http://www.springframework.org/schema/beans"... >  
    <bean id="saludator"  
        class="com.openwebinars.beans.Saludator">  
        <constructor-arg name="str"  
            value="Hola alumnos de openwebinars"></constructor-arg>  
        </bean>  
</beans>
```

# UN BEAN QUE REFERENCIA A OTRO BEAN

- ▶ Como cabe suponer, se puede inyectar dependencias más allá de tipos primitivos.

```
<beans xmlns="http://www.springframework.org/schema/beans"... >
  <bean id="emailService"
        class="com.openwebinars.beans.EmailService">
    <property name="saludator" ref="saludator"></property>
  </bean>

  <bean id="saludator" class="com.openwebinars.beans.Saludator">
    <property name="mensaje"
              value="Hola alumnos de openwebinars"></property>
  </bean>
</beans>
```

# BEANS QUE IMPLEMENTAN INTERFACES

```
public interface IEmailService {  
    public void enviarEmailSaludo(String str);  
}  
  
public class EmailService implements IEmailService{  
    //...  
}  
  
public class App {  
    public static void main(String[] args) {  
        //...  
        IEmailService emailService = null;  
        emailService = appContext.getBean(IEmailService.class);  
        emailService.enviarEmailSaludo("luismi@openwebinars.net");  
        //...  
    }  
}
```

# OTROS ASPECTOS DE LOS BEANS QUE NOS DEJAREMOS EN EL TINTERO

- ▶ Inner beans (beans anidados)
  - ▷ Definimos un bean dentro de otro.
  - ▷ Similar a las referencias.
  - ▷ Ámbito más restrictivo.
- ▶ Colecciones
  - ▷ Posibilidad de inyectar valores de una colección
  - ▷ `<list>`, `<set>`, `<map>`, `<props>`