

# INYECCIÓN AUTOMÁTICA O AUTOCABLEADO



# INYECCIÓN AUTOMÁTICA

- ▶ Spring permite la inyección *automática* entre beans que *se necesitan*.
- ▶ Busca candidatos dentro del contexto.
- ▶ Ventajas
  - ▷ Reduce la configuración necesaria
  - ▷ Útil durante el desarrollo. Permite requerir objetos sin configurarlo explícitamente.

# INYECCIÓN AUTOMÁTICA

## AUTOWIRED

```
<beans xmlns="http://www.springframework.org/schema/beans" ...>

    <bean id="emailService"
          class="com.openwebinars.beans.EmailService"
          autowire="byType">

    </bean>

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

## TIPOS DE AUTOWIRED

- ▶ **no**: sin autocableado
- ▶ **byName**: en función del nombre de la propiedad requerida.
- ▶ **byType**: en función del tipo de la propiedad requerida. Si hay más de un bean de este tipo, se produce excepción.
- ▶ **constructor**: análogo a *byType*, pero para argumentos del constructor.

# INCONVENIENTES DEL AUTOCABLEADO

- ▶ Es útil si se usa siempre en un proyecto.
- ▶ En otro caso, puede ser confuso.
- ▶ No se pueden *autoinyectar* tipos primitivos o String.
- ▶ Menos exacto que la inyección explícita
- ▶ Posible ambigüedad en inyección *byType*.

# INCONVENIENTES DEL **AUTOCABLEADO**: ¿QUÉ HACER?

- ▶ No usar el autocableado :(
- ▶ Manejar el autocableado a través de anotaciones (lo estudiaremos más adelante).
- ▶ Utilizar `autowired-candidate=false` en los beans más conflictivos.
- ▶ Utilizar `primary=true` en las opciones principales.

# AUTOWIRED-CANDIDATE

- ▶ Nos permite excluir a un bean de ser autoinyectado.
- ▶ Se sigue permitiendo que se pueda inyectar de forma explícita.

```
<beans xmlns="http://www.springframework.org/schema/beans" ...>

    <bean id="saludator"
          class="com.openwebinars.beans.Saludator"
          autowire-candidate="false" >
        ...
    </bean>
</beans>
```

# PRIMARY

- ▶ Nos permite indicar que un bean de un tipo tendrá preferencia sobre los demás del mismo tipo.

```
<beans xmlns="http://www.springframework.org/schema/beans" ...>
  <bean id="emailService"
        class="com.openwebinars.beans.EmailService"
        autowire="byType">
  </bean>
  <bean id="saludator"
        class="com.openwebinars.beans.Saludator"
        primary="true" >
    ...
  </bean>
  <bean id="englishSaludator"
        class="com.openwebinars.beans.Saludator">
    ...
  </bean>
</beans>
```