

ESTRUCTURA DE PROYECTO JAVA + MAVEN

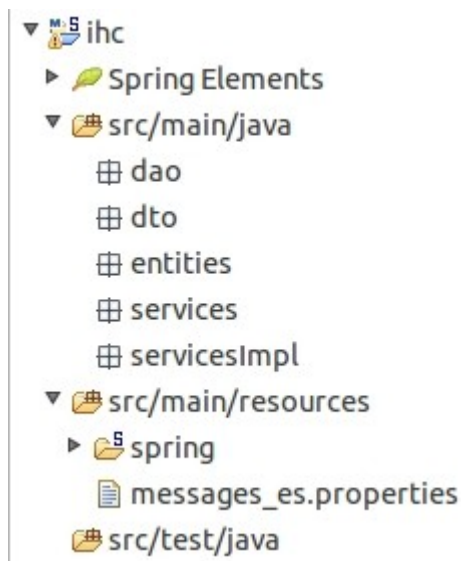
El subdirectorio '`src/main/java`' contendrá nuestras clases java fuente.

- **dto: Objeto de Transferencia de Datos** : son objetos simples que no deben contener lógica de negocio que requiera pruebas
- **dao: Objeto de Acceso a Datos** : El objeto de acceso a datos es básicamente un objeto o una interfaz que proporciona acceso a una base de datos o cualquier otro medio de almacenamiento de persistencia
- **entities**: Es un **objeto de dominio persistencia**, representa una tabla de una base de datos relacional.
- **services**: Un **servicio** es un componente que tiene lógica de negocio que se puede reutilizar fácilmente en cualquier otra parte de la aplicación, sin importar dónde se encuentra el servicio, o lo que el mecanismo de comunicación se utiliza entre el cliente y el servicio.
- **servicesImpl**: Corresponde a la implementación del **servicio** y será la capa disponible hacia la webapp.

El subdirectorio '`src/main/resources`' contendrá todos los recursos utilizados por la aplicación.

- **messages_es.properties**: archivo contenedor de mensajes de control de errores .

El subdirectorio '`src/test/java`' contendrá las clases de test que se encargarán de probar el correcto funcionamiento de nuestra aplicación. Ejemplo: test unitarios de **Junit**.

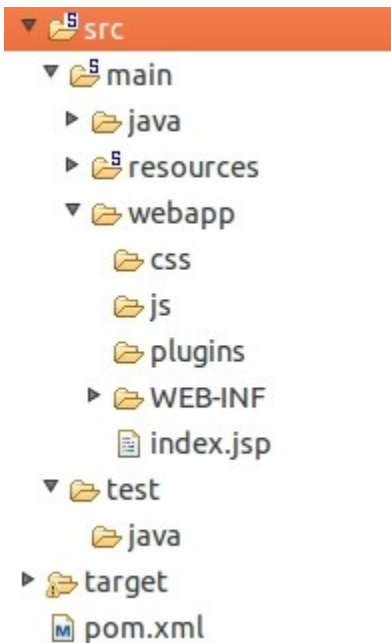


El subdirectorio '**src/main/webapp**' alojará todos los archivos que no sean código fuente Java, como archivos JSP.

- **css**: Contenedor de estilos
- **js**: Contenedor archivos javascript
- **plugins**:
 - Validation.Engine: valida formato de input de texto
 - JQuery:

El directorio '**target**' donde se generará el archivo **WAR** que usaremos para almacenar y desplegar rápidamente nuestra aplicación.

El fichero '**pom.xml**' contiene las dependencias Maven.



Aplicar estándares que aparecen en los siguientes link:

<http://javafoundations.blogspot.com/2010/07/java-estandares-de-programacion.html>

JavaDoc

Clases e interfaces

Los nombres de clases deben ser sustantivos y deben tener la primera letra en mayúsculas. Si el nombre es compuesto, cada palabra componente deberá comenzar con mayúsculas.

Los nombres serán simples y descriptivos. Debe evitarse el uso de acrónimos o abreviaturas, salvo en aquellos casos en los que dicha abreviatura sea más utilizada que la palabra que representa (URL,

HTTP, etc.).

Las interfaces se nombrarán siguiendo los mismos criterios que los indicados para las clases. Como norma general toda interfaz se nombrará con el prefijo "I" para diferenciarla de la clase que la implementa (que tendrá el mismo nombre sin el prefijo "I").

```
class Ciudadano
class OrganigramaDAO
class AgendaService
class IAgendaService
```