**Separación de funcionalidades**

Conviene programar usando interfaces, de manera que sus implementaciones pueden variar sin modificar el programa que las esté utilizando. Para instanciar estos objetos se utiliza el patrón “Factory”, con lo que se consigue desacoplar el código.

Mediante los interfaces “provider” se desacoplan las implementaciones de los distintos módulos del programa, logrando mayor flexibilidad.

ApplicationContext es una interfaz de Spring que permite instanciar de manera transparente los “beans” que hayan sido declarados, bien en el xml de configuración (app-context.xml), bien mediante anotaciones.

IMPORTANTE: Ver los “relative paths” en aplicaciones web Java.

**CURSO DE MIGUEL ÁNGEL**

SPRING:

**Anotaciones vistas hasta ahora:**

- @Component (“beans” de Spring)

@Controller

@ControllerAdvice

@Configuration

- @Mapping

@RequestMapping

@GetMapping

@PostMapping

- @RequestParam

- @PathVariable

- @ExceptionHandler

- @Autowired

- @Qualifier

**Interfaces vistas hasta ahora:**

- org.springframework.ui.Model

- org.springframework.web.servlet.HandlerInterceptor

- org.springframework.web.servlet.config.annotation.WebMvcConfigurer

**Clases vistas hasta ahora:**

- org.springframework.web.servlet.ModelAndView

**Clases abstractas vistas hasta ahora:**

**-** org.springframework.web.servlet.handler.HandlerInterceptorAdapter (ver la interfaz HandlerInterceptor)

- org.springframework.web.servlet.config.annotation.WebMvcConfigurerAdapter (ver interfaz WebMvcConfigurer)

SPRING BOOT:

Por defecto, SpringBoot buscará las páginas de error en “templates/error”.

THYMELEAF:

Cuando Thymeleaf analiza un .html, no sabe resolver las rutas en la propiedad “src”. Hay que indicarlas con *th:src=”@{/imgs/2.png}”* (Spring busca por defecto los recursos estáticos en la carpeta “static”, de manera que es la ruta base sobre la que trabaja al poner “@{}”).