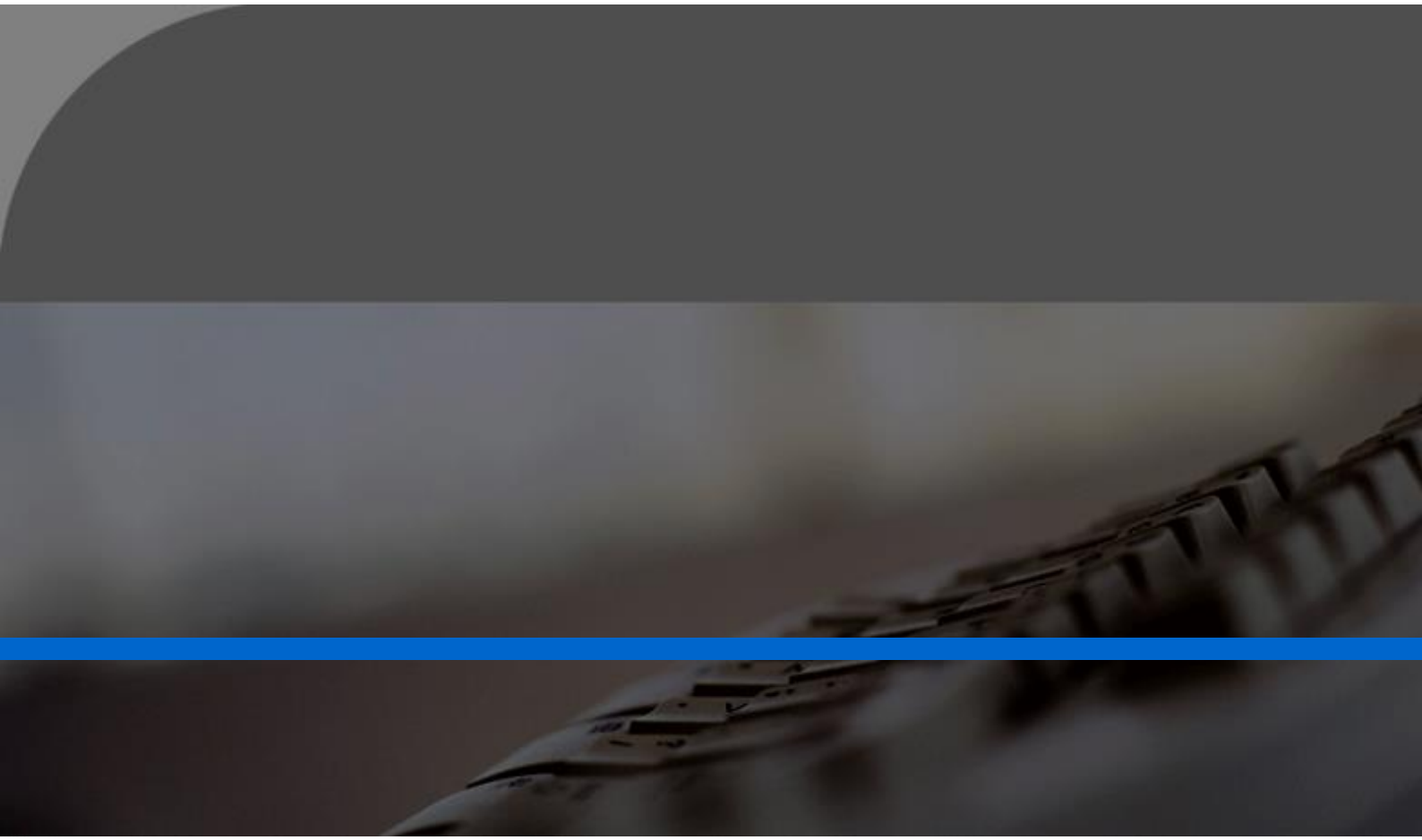
GOBIERNO de CANTABRIA
Consejería de Presidencia y Justicia



3 de Junio de 2019

Versión 1.3.4

APLICACIÓN AMAP 2.1

CONVENCIONES DE CÓDIGO EN DESARROLLO JEE	
Todas los ficheros están codificados en UTF-8	
Se le ha asignado a la aplicación un código identificativo único	
Sigue la estructura de directorios especificada	
ESTANDAR DE CODIFICACIÓN JAVA (no aplicable a código fuente generado automáticamente)	
NOMENCLATURA – Generalidades	
El idioma por defecto a la hora de dar sentido funcional al nombre de clases, variables, constantes, etc. es una mezcla entre la nomenclatura tradicional en inglés y la nomenclatura funcional adoptada.	
NOMENCLATURA - Paquetes	
El paquete base está definido como *es.gobcantabria.aplicaciones.<ID_APP> para aplicaciones, es.gobcantabria.amap.<ID_GRP>.<ID_APP> para componentes AMAP y es.gobcantabria.trewa.<ID_APP> para procedimientos Trew@*	
Los nombres de todos los paquetes están escritos en minúsculas y sin caracteres especiales.	
No existe ninguna clase en el paquete base	
La estructura de paquetes sigue la estructura definida (ver documento)	
NOMENCLATURA - Interfaces	
Todos los nombres de los interfaces utilizan el sufijo *Interface*	
Todos los nombres de los inferfaces están escritos en formato *CamelCase*	
No se usan abreviaciones que dificultan la compresión del código	
NOMENCLATURA – Clases	
Los nombres están escritos en formato *CamelCase*	
Los nombres son simples y descriptivos.	
Se usan palabras completas sin acrónimos y abreviaturas	
NOMENCLATURA – Gestiones	
Se emplea la nomenclatura *<<FuncionalidadGenerica>><<Entidad>><<Especificación de Clase>>*	
NOMENCLATURA – Métodos	
Los métodos son verbos en infinitivo	
Están en formato *lowerCamelCase*	

No contienen caracteres especiales	
Los nombres son suficientemente descriptivos.	
NOMENCLATURA – Variables	
Están en formato *lowerCamelCase* (no aplicable a variables finales)	
No contienen caracteres especiales	
Los nombres son suficientemente descriptivos.	
ESTILO DE CODIFICACIÓN – Comentarios	
Se evitan referencias al diseño funcional	
No se hace un uso abusivo de ellos	
Se evita el uso de caracteres especiales y "gráficos" en ASCII	
ESTILO DE CODIFICACIÓN – JavaDoc	
Se proporciona el *Javadoc* de cada clase/interface, método, propiedad o constante creada.	
Se escribe siempre en tercera persona	
Los caracteres especiales como tildes, eñes, etc se codifican con su código HTML correspondiente o bien en UTF-8.	
ESTILO DE CODIFICACIÓN – Declaraciones	
Se utiliza una declaración de cada vez.	
Se inicializan todas las variables locales (excepto si son propiedades de un bean)	
Las variables de avance de bucles `for` no son modificadas fuera de la propia sentencia del bucle.	
Se evita la duplicidad de los nombres de variables en diferentes niveles dentro de la misma clase.	
ESTILO DE CODIFICACIÓN – Sentencias	
No hay más de una sentencia por línea de código	
Todo bloque de sentencias se encuentra entre llaves.	
Se definen las tres condiciones del bucle `for`	
La variable de avance del bucle nunca es modificada dentro del propio bucle.	
BUENAS PRÁCTICAS – Constantes	
Ninguna constante numérica se codifica directamente.	
BUENAS PRÁCTICAS – Propiedades	
El acceso/modificación de las propiedades de una clase (no constantes) siempre mediante métodos de acceso `get/set`.	

La asignación de variables / propiedades no es consecutiva.	
No se utiliza el operador de asignación en sitios donde se puede confundir con el operador igualdad ni dentro de expresiones complejas.	
BUENAS PRÁCTICAS – Métodos	
No se accede a un método estático desde una instancia de una clase.	
NOMENCLATURA	
Los nombres de los ficheros JSP siguen la notación lowerCamelCase	
CÓDIGO JSP/HTML	
No se emplean scriptlets.	
No se incluyen includes dinámicos	
Los atributos de los tag HTML van entre comillas dobles	
No se utiliza Javascript para la creación de contenido	
No se utilizan elementos ni atributos HTML deprecated (html 4)	
Se usa CSS para aplicar los estilos	
Se evita el uso de comentarios en HTML	
Todos los literales están internacionalizados	
OTRAS CONSIDERACIONES	
FICHERO DE LOG	
Se especifica el logging-profile en el fichero MANIFEST.MF (no aplicable para procedimientos trew@)	
FICHERO DE PROPIEDADES	
Las propiedades relacionadas con sistemas se guardan en un fichero de propiedades externo a la aplicación	
La nomenclatura del fichero es la adecuada.	
La nomenclatura de las propiedades es la adecuada.	
Normalización de variables de los ficheros de propiedades.	
LIBRERÍAS Y FRAMEWORKS	
Se utilizan las librerías especificadas en el FMW AMAP 2.0 (para procedimientos trew@ se permite además el uso del repositorio "amap-trewa"). Quedan excluidas las aplicaciones del FMW AMAP 1.5	
DATASOURCES	
El datasource se define vía jndi	
La variable jndi sigue la nomenclatura especificada	
VERSIONADO	

El software entregado especifica un número de versión y se corresponde con el versionado del código fuente. Esta versión deberá ser posterior a la del último despliegue en producción.	
En el pom.xml principal se indicará en una propiedad la versión del arquetipo que se ha utilizado. Esta propiedad es generada automáticamente al crear un proyecto con el arquetipo y no deberá ser alterada ni modificada.	
EMPAQUETADO	
El nombre del distribuible sigue la nomenclatura especificada.	
El nombre del contexto web debe coincidir con el nombre de la aplicación o en su defecto, estar notificado y registrado en el inventario de aplicaciones.	
PRUEBAS UNITARIAS	
El software debe tener y ejecutar correctamente sus pruebas unitarias.	
El proyecto debe incluir JaCoCo para obtener la cobertura de los tests	
CÓDIGO FUENTE	
Se ha proporcionado el código fuente	
El ear proporcionado coincide con el ear generado desde el código fuente.	
DOCUMENTACIÓN	
Existencia de DRF y Análisis con información coherente como indican las normas de AMAP	
Existencia de documentación de pruebas y Manual de usuario como indican las normas de AMAP (no necesario si no es despliegue de producción)	
En el Inventario de Aplicaciones (INVAPP) se debe indicar todos los componentes amap utilizados	
CONSULTAS A BASE DE DATOS	
Las consultas serán de un rendimiento razonable, en caso de requerirse consultas que requieran de una cantidad masiva de registros o con una mezcla de tablas poco convencional (no unida por claves ajenas, campos indexados o similares) deberán ser indicadas al grupo de arquitectura para su validación.	
VERSIÓN DE LOS COMPONENTES AMAP	
Los componentes AMAP empleados en las aplicaciones que estén recogidos en el POM padre, no deberán indicar la versión. Para los componente no recogidos en el POM padre, será recomendable el	

uso de un rango para indicar la versión.	
AMAP-GESTOR-DOCUMENTAL	
Se debe utilizar el gestor documental versión 2.x (indicar en el resumen el incumplimiento)	
SEGURIDAD	
INYECCIÓN SQL	
No deben existir generación de consultas SQL basada en la concatenación directa (sin comprobación) de parámetros obtenidos de la petición. Utilizar en su lugar procedimientos preparados con variables parametrizadas.	
CROSS-SITE SCRIPTING (XSS)	
No deben existir generación de elementos de presentación (html y javascript especialmente) basada en la utilización directa (sin comprobación) de parámetros obtenidos de la petición.	

* NOTA: las normas en negrita son bloqueantes

1..1. **RESUMEN**