



# Anexo VI

## Directrices de desarrollo

### *Documento de Estándares*

## 2012

Revisión	Fecha
1	Marzo, 2012

## Índice

Estándares de desarrollo .....	4
Alcance .....	4
Estructura .....	6
Procesos organizativos .....	6
Procesos operativos .....	6
Flujo de Procesos .....	7
Requerimientos de referencia .....	8
Arquitectura .....	8
Contextos de albergue .....	8
Normativas y documentación .....	9
Herramientas de desarrollo .....	9
Sistemas horizontales .....	9
Estándares de calidad de sistemas software .....	10
Introducción .....	10
Objetivo .....	10
Estándares aplicados .....	11
Oficina Técnica de Calidad de producto y pruebas .....	12
Modelo de Aseguramiento de la Calidad .....	12
Asignación del valor del NAC .....	14
Plan SQA y actividades de aseguramiento de la calidad .....	14
Verificación documental .....	15
Auditorías Fin de Fase SQA .....	16
Auditoría Final SQA .....	16
Indicadores .....	17
Metodología de desarrollo ARINbide .....	18
Alcance .....	18
Ingeniería del Software (ISW) .....	19
Catálogo de Requisitos de Usuario .....	19
Análisis del Sistema de Información (ASI) .....	20
Diseño del Sistema de Información (DSI) .....	21
Construcción del Sistema de Información (CSI) .....	22
Implantación y Aceptación del Sistema (IAS) .....	22
Metodología de pruebas Probamet .....	24
Alcance .....	24
Metodología .....	24
Planificación de las Pruebas (PPB1) .....	24
Seguimiento de las Pruebas (PPB2) .....	25
Análisis y Diseño de las Pruebas (APB) .....	26
Ejecución de las Pruebas (EPB) .....	27
Utilidades de Desarrollo de Aplicaciones (UDA) .....	28
Introducción .....	28
Arquitectura Tecnológica .....	28
Patrones de interacción .....	29

Generadores de código..... 30  
Kit de desarrollo..... 31

# Estándares de desarrollo

## Alcance

Desde una perspectiva tecnológica, es conveniente que todas las actividades y tareas necesarias en el proceso de desarrollo del software sean estandarizadas e interconectadas entre sí, facilitando y acelerando de este modo, ya no sólo dicho proceso, sino también la propia operativa de creación de los sistemas.

Es objeto de la presente área de estandarización:

- Establecer el conjunto mínimo de requerimientos y recomendaciones técnicas que estandaricen el proceso de desarrollo de software en las fases definidas por la metodología de aplicación
- Definir una serie de instrucciones de trabajo estandarizadas y coherentes en dicho proceso.
- Proporcionar un marco de referencia de terminología y vocabulario común para el desarrollo del software
- Identificar las principales expectativas a gestionar en las distintas modalidades de contratación del desarrollo. Con recursos humanos y/o materiales propios y/o ajenos, como adquisición de “producto”, etc. Se fijará especialmente en los entregables (código, ejecutables, documentación, etc.) a exigir y auditar
- Estandarizar las herramientas de gestión asociadas a las fases del ciclo de vida del proceso de desarrollo del software
- Enfatizar las necesidades de gestión de la calidad de los productos a implantar orientadas a minimizar al máximo los fallos del servicio en el entorno de producción
- Limitar las posibles arquitecturas software y hardware, ajustándolas al modelo de albergue de GV-EJIE

El estándar contiene un conjunto de procesos, actividades y tareas diseñadas para ser utilizadas en los proyectos de desarrollo software, alineadas en todos los casos con la metodología normalizada, así como el conjunto de roles identificados y los que intervendrán en cada uno de dichos procesos, actividades y tareas.

Para cada proceso, actividad y tarea en que se descompone el estándar se indica el grado de exigencia para su conformidad:

Indicador	Descripción
<ul style="list-style-type: none"> <li>• <b>Deberá</b></li> </ul>	Indica un requisito obligatorio para su conformidad
<ul style="list-style-type: none"> <li>• <b>Debería</b></li> </ul>	Indica una fuerte recomendación que no es obligatoria para su conformidad
<ul style="list-style-type: none"> <li>• <b>Puede</b></li> </ul>	Indica una forma autorizada de cumplir un requisito o de evitar la necesidad de satisfacer la conformidad
<ul style="list-style-type: none"> <li>• <b>Si...entonces</b>   <b>deberá</b> /  <b>debería</b> /  <b>puede</b></li> </ul>	Indica que el grado de conformidad indicado está sujeto al cumplimiento de ciertas condiciones



Del mismo modo, se remarcan aquellos aspectos que deben ser tratados y cumplidos de manera rigurosa

## Estructura

La Guía de estándares de desarrollo de sistemas software se descompone en:

### Procesos organizativos

Engloba los procesos más relacionados con la gestión y preparación, necesarios para el desarrollo del sistema software.

Proceso	Descripción
Preparación y provisión	Definir el alcance del desarrollo a realizar, las infraestructuras técnicas necesarias para su ejecución y los acuerdos de gestión del proceso, comprendiendo las actividades de: Adquisición, Equipamiento e Infraestructura técnica
Administración del proyecto	Comprende las actividades y tareas necesarias para la correcta administración del proyecto, en lo referente a Gestión del Proyecto y Gestión del Cambio

### Procesos operativos

Define los procesos a realizar desde el punto de vista de la construcción del sistema software.

Proceso	Descripción
Análisis	Comprende las actividades y tareas dictadas por la metodología de aplicación ARINbide en la fase de <i>análisis del sistema de información</i> (ASI), las definidas por Probamet en la fase de <i>planificación de pruebas</i> (PPB1) y parcialmente, en la fase de <i>análisis y diseño de las pruebas</i> (APB)
Diseño	Comprende las actividades y tareas dictadas por la metodología de aplicación ARINbide en la fase de <i>diseño del sistema de información</i> (DSI), y parcialmente, las definidas por Probamet en la fase de <i>análisis y diseño de las pruebas</i> (APB)
Implementación	Comprende las actividades y tareas dictadas por la metodología de aplicación ARINbide en la fase de <i>construcción del sistema de información</i> (CSI), y parcialmente, las definidas por Probamet en la fase de <i>ejecución de pruebas</i> (EPB). Se indican también las mejores prácticas de aplicación en dicha fase, orientadas a mejorar tanto la ejecución del propio proceso, como a optimizar la tarea de entrega del sistema software
Implantación y aceptación del sistema	Comprende las actividades y tareas dictadas por la metodología de aplicación ARINbide en la fase de <i>implantación y aceptación del sistema</i> (IAS), y parcialmente, las definidas por Probamet en la fase de <i>ejecución de pruebas</i> (EPB)
Aseguramiento de la calidad	Asegurará que los productos obtenidos y los procesos del ciclo de vida del proyecto cumplen con los requerimientos y los planes establecidos, mediante la aplicación Modelo SQA

Para cada una de ellos se indican además las herramientas de uso en cada actividad y tarea en las que se descomponen.

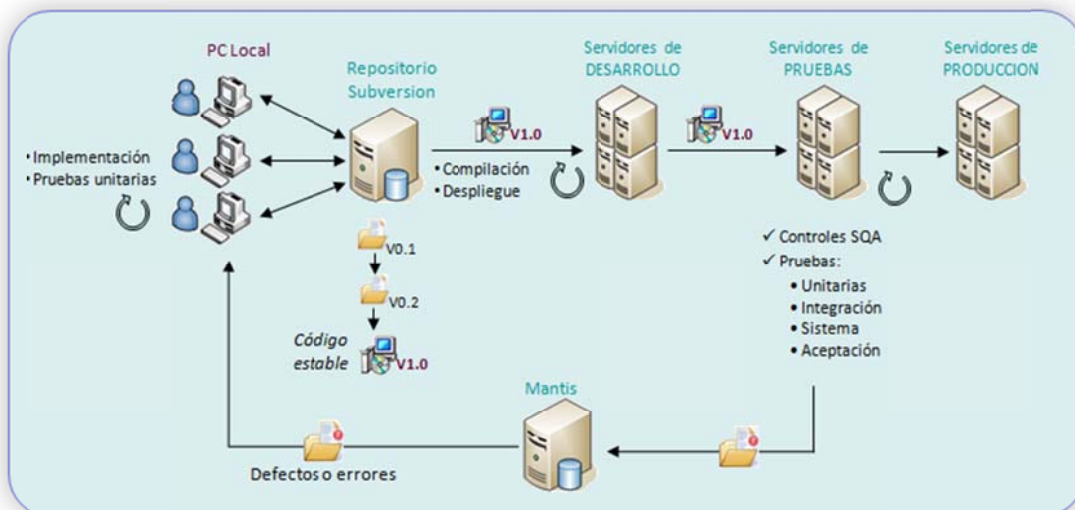
## Flujo de Procesos

La Guía de estándares de desarrollo de sistemas software establece también el flujo de ejecución de procesos que **deberá** seguir el proceso de desarrollo de un sistema software, detallando el conjunto de actividades y tareas necesarias, así como los roles involucrados.

El flujo de ejecución de procesos **deberá** seguir el siguiente esquema:



Y para el proceso de implementación:



## Requerimientos de referencia

Con objeto de contextualizar la Guía de estándares de desarrollo de sistemas software dentro del entorno de albergue de los sistemas software a construir, se facilitan un conjunto de requerimientos básicos de referencia que deberán tenerse en consideración:

### Arquitectura

La arquitectura física de los sistemas software se ve afectada por el modelo de seguridad corporativo definido. Y, por ende, su arquitectura lógica se ve segmentada también en tres —o más— niveles con cometidos distintos: uno para los servicios de usuario —presentación—, otro para los servicios de la lógica de negocio —la lógica principal de la aplicación—, y otro nivel para los servicios de datos.

Existe además una capa de integración, que provee el modelo de compartición de funciones y datos y orquestación de servicios entre distintos sistemas, departamentales y horizontales.

Se identifican, además, distintos entornos operativos:

Entorno	Descripción
Desarrollo	Sobre el que se realizará la primera integración y prueba de lo desarrollado y testeado en PC local, simulando la arquitectura y el entorno tecnológico final que sustentará el aplicativo
Pruebas	Sobre el que se llevarán a cabo las principales tareas de aseguramiento de la calidad, se realizarán los ensayos de instalación del aplicativo y las pruebas de aceptación del usuario
Producción	Entorno productivo final en el que se ejecutará la aplicación con los usuarios finales y los datos reales

El traspaso entre los diferentes entornos sucesivos se realizará mediante la sistemática de traspasos ya normalizada.

### Contextos de albergue

Cada sistema software se deberá ejecutar en alguno de los diferentes contextos de albergue existentes según sus características de negocio y por tanto de las tipologías de usuarios a los que da soporte:



Contexto	Descripción
Internet	Para aplicaciones que requieren accesos desde Internet. Este contexto tiene connotaciones especiales en lo referente a la publicación de los contenidos estáticos y estilos
Intranet	Para aplicaciones que requieren únicamente accesos de usuarios corporativos del Gobierno Vasco
Extranet	Aplicaciones que requieren exponer todo o parte de su negocio a un grupo acotado de usuarios —generalmente, adscritos a otras administraciones u organizaciones afines— apoyándose en otras redes que no se corresponden con la intranet de GV. Actualmente, existen dos tipos de extranet: la de «jakinaplus» y la de «JASO»

## Normativas y documentación

Para asegurar la gobernabilidad de los sistemas software y, por tanto, de sus procesos de instalación y albergue en los servidores corporativos, se dispone de un conjunto de normativas y documentación adicional. Éstos se asocian tanto a tecnologías como a productos, como a los distintos sistemas básicos de infraestructura:

- JAVA: Normas de albergue de aplicaciones «*Weblogic*», guías de desarrollo, estándares, librerías y productos
- .NET: Normas de albergue .NET, desarrollo e implantación
- ORACLE: Estándares y normas de uso de Oracle
- PLATEA—Presencia en Internet: Normativas, guías de desarrollo y documentación técnica para el uso de la infraestructura de Presencia en Internet
- PLATEA—Tramitación: Guías de uso de la infraestructura de tramitación
- Otros sistemas corporativos: Normativas de desarrollo y guías de procedimiento de aplicaciones horizontales y sistemas de uso común: Dokusi, Pasarela de Pagos, K31/O75, XLNets, SMS Corporativo, GIS, Nora

## Herramientas de desarrollo

Se enumeran el conjunto de herramientas homologadas para dar soporte al ciclo completo de vida de desarrollo de los sistemas software. Éstas, además, se ven referenciadas y marcadas de uso obligatorio en la misma Guía de estándares de desarrollo de sistemas software.

## Sistemas horizontales

Se enumeran los sistemas básicos de infraestructura sobre los cuales deben generarse los nuevos aplicativos software a construir: XLNetS, PLATEA, Utilidades de Desarrollo de Aplicaciones (UDA), Control M, K31/O75, Pasarela de pagos, Nora, SGA (Sistema de Gestión de Archivos), Dokusi.

# Estándares de calidad de sistemas software

## Introducción

Como metodologías de desarrollo de sistemas software, tanto Métrica3 como ARINbide —en su versión inicial—, ya establecen las actividades y tareas de pruebas de los sistemas a construir, pese a lo cual, en algunos casos, éstas no se definen adecuadamente o bien no se les otorga la importancia que merecen. Hay que considerar siempre que las eventualidades producidas a raíz de fallos en los sistemas, además de provocar un coste económico importante, principalmente por el elevado número de personas involucradas en su resolución, también producen la pérdida de confianza de los usuarios en el sistema.

En un escenario en el que los sistemas software se desarrollan y construyen por terceros proveedores, el contratante del servicio, como primer receptor del mismo, en muchos casos debe confiar en el buen hacer del proveedor seleccionado, especialmente si no dispone de los medios apropiados para auditar la entrega y en su caso argumentar defectos en el proceso de desarrollo. En general, una vez validado que el sistema responde a los principales requisitos funcionales especificados, el usuario realizará las pruebas de aceptación, corrigiéndose los errores encontrados y traspasándose al fin al entorno de producción. Sin embargo, se observa necesario en muchos casos la validación rigurosa de los requisitos funcionales y los no funcionales, y las validaciones que aseguren que el sistema es lo suficientemente robusto y estable como para pasar a un entorno productivo con las garantías adecuadas.

Todo sistema software debe ser construido con el fin de ser explotado de manera satisfactoria por parte de los usuarios finales, por lo que dicho sistema deberá presumir de la calidad necesaria para poder asegurarlo. La calidad en este ámbito se define precisamente como el grado con el que un sistema, componente o proceso cumple los requerimientos especificados y las necesidades o expectativas del cliente o usuario.

Dentro del concepto calidad software se puede distinguir entre calidad del proceso y la calidad del producto. La calidad de los productos software viene dada por los procesos que lo desarrollan, aunque para los usuarios finales, la calidad es una característica propia del producto.

## Objetivo

El propósito de la presente área de estandarización se centra por tanto en el aseguramiento de la calidad del producto. Para ello se establece el modelo básico de aseguramiento de la calidad de los productos software que se deban implantar en el entorno de GV-EJIE.

El Modelo de Aseguramiento de Calidad de Sistemas Software o Modelo SQA, es el marco de referencia que engloba todas las actividades relacionadas con el aseguramiento de calidad durante todo el ciclo de vida de desarrollo y pruebas.

## Estándares aplicados

El Modelo SQA definido, se apoya y toma como referencia una serie de estándares comúnmente reconocidos (**ISO** e **IEEE**):

- Se adapta a la Norma UNE-EN ISO 9001:2000 Sistema de Gestión de Calidad, en lo referente al enfoque que propone de gestión de calidad basado en procesos interrelacionados y los requisitos de cliente con el fin de lograr la satisfacción del cliente mediante la aplicación efectiva del sistema.
- Toma como referencia el estándar ISO/IEC 9126:1991 Ingeniería del Software – Calidad de Producto, la cual contiene un modelo de calidad y medición que permite la evaluación de la calidad de un producto software, y las series SQuaRE ISO/IEC 25000. El modelo SQuaRE (Software Product Quality Requirements and Evaluation) es una revisión de la norma ISO/IEC 9126-1:2001.
- IEEE Std.730 – 2002: IEEE Standard for Software Quality Assurance Plans. Define la información que debe contener un plan de aseguramiento de la calidad software, y su relación con otros procesos implicados (gestión de incidencias, gestión de la configuración).
- IEEE Std.829 – 1998: IEEE Standard for Software Test Documentation. Define la documentación generada en cada una de las fases del proyecto de pruebas.
- IEEE Std.830 – 1998: IEEE Recommended Practice for Software Requirements Specifications. Proporciona una guía de buenas prácticas para la elaboración de una especificación de requisitos.
- IEEE Std.1012 – 2004: IEEE Standard for Software Verification and Validation. Detalla los procesos de verificación y validación (V&V) del software, y su organización.
- IEEE Std.1061 – 1998: IEEE Standard for a Software Quality Metrics Methodology. Define el establecimiento, la implementación, el análisis y la validación de métricas de calidad de software.

El Modelo SQA se basa también en las buenas prácticas del Modelo **CMMI** «*Capability Maturity Model Integration*». Una de las variantes del modelo, es el **CMMI-ACQ** «*CMMI for Acquisition*» que está orientado a la gestión de adquisiciones para organizaciones con un porcentaje elevado de subcontratación de desarrollos de software. El Modelo CMMI-ACQ se estructura en diferentes



Áreas de Proceso relativas a la Gestión de Proyectos, la Gestión de Procesos, la Gestión de la Adquisición y otros procesos de Soporte. Es un Modelo estructurado en 5 niveles de madurez, cada uno de los cuales contiene un conjunto de buenas prácticas agrupadas en áreas de proceso.

Para la definición del Modelo SQA y la metodología de pruebas PROBAMET se han tenido en cuenta ciertas pautas del Modelo CMMI-ACQ, principalmente las Áreas de Proceso de los niveles de madurez 2 y 3, como son:

1. Enfoque en el proceso organizativo OPF (nivel 3)
2. Definición de proceso organizativo OPD (nivel 3)
3. Gestión Técnica de la Adquisición ATM (nivel 3)
4. Validación de la Adquisición AVAL (nivel 3)
5. Verificación de la Adquisición AVER (nivel 3)
6. Medición y Análisis MA (nivel 2)
7. Aseguramiento de la Calidad PPQA (nivel 2)
8. Gestión de la Configuración CM (nivel 2)

## Oficina Técnica de Calidad de producto y pruebas

Según el estándar IEEE 1012, se recomienda que el control de calidad de software sea realizado por un equipo experto e independiente al grupo de desarrollo, trabajando paralelamente junto con éste durante el ciclo de vida de desarrollo pero gestionado de forma autónoma, garantizando así su independencia.

En función del volumen de la contratación, de la criticidad del servicio que se desea resolver, del posible coste adicional al propio proyecto de desarrollo, o de cualquier otro criterio que se considere relevante, el contratista  **debería** estimar conveniente la contratación de una Oficina Técnica de Calidad (OTC) de producto y de pruebas que permita asegurar el nivel de calidad de producto adecuado a las expectativas establecidas para el servicio, en cuyo caso se deberá seguir la metodología de calidad definida para tal efecto. No obstante, y con independencia de la contratación o no de dicha oficina, se  **deberá** asegurar que el producto software obtenido cumple el estándar de calidad detallado en el presente documento.

## Modelo de Aseguramiento de la Calidad

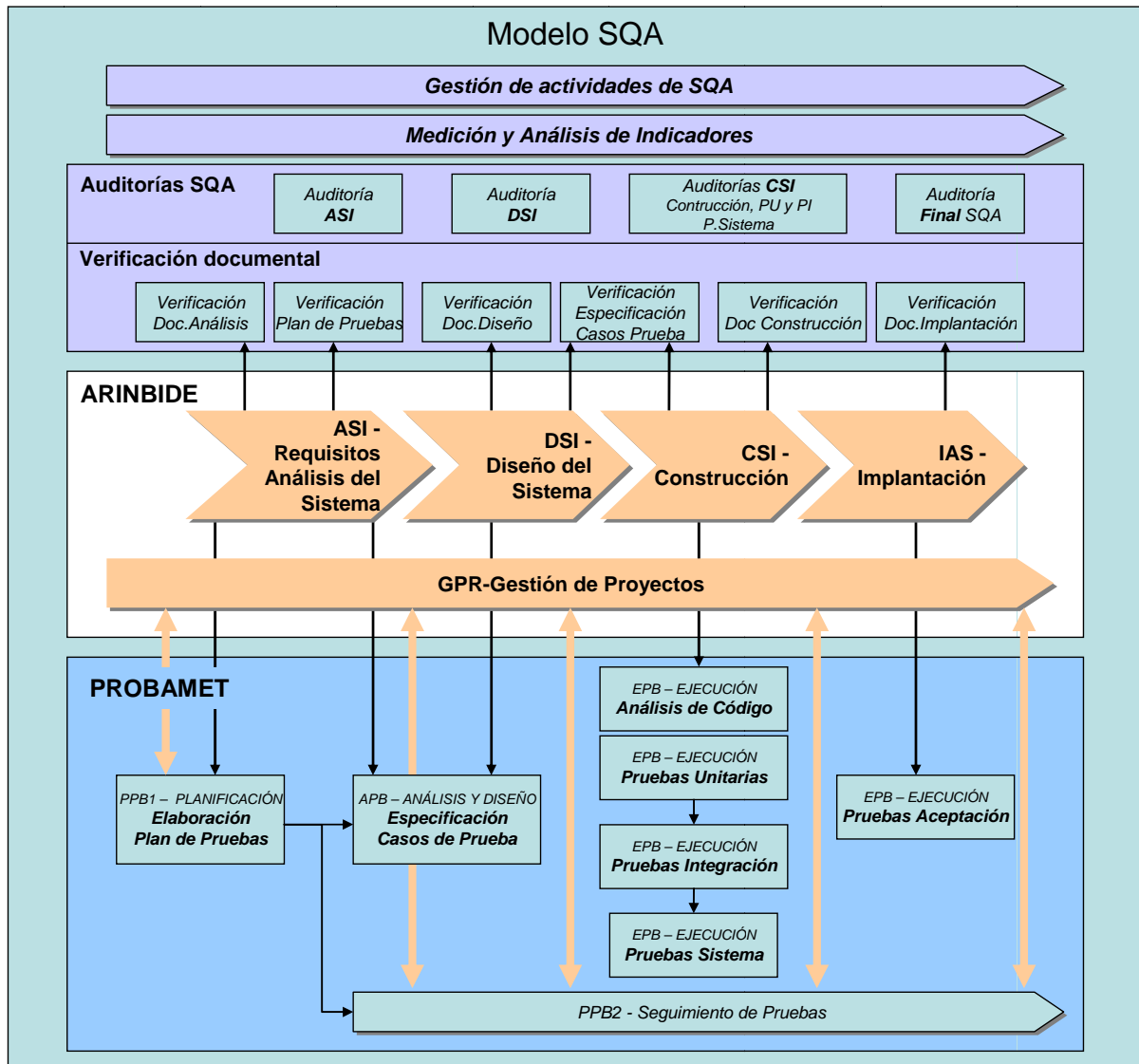
A grandes rasgos, el modelo de aseguramiento de la calidad del producto software – Modelo SQA - consta de:

1. Asignación del valor NAC (nivel de aseguramiento de la calidad) asociado al proyecto

2. Elaboración del Plan SQA del proyecto, definiendo las actividades de aseguramiento de calidad a realizar durante el ciclo de vida dependiendo del NAC asociado al proyecto
3. El proceso para la realización de las actividades de aseguramiento de calidad definidas, alineadas con la metodología de desarrollo ARINbide y cumpliendo implícitamente la metodología de pruebas Probamet
4. Los indicadores de calidad estándar y sus umbrales permitidos
5. El conjunto de herramientas que facilitan la aplicación del modelo y las metodologías

El contratista **deberá** incluir en el Pliego de Bases Técnicas los detalles suficientes relativos a la aplicación del modelo de aseguramiento de la calidad para que el licitador pueda ofertar adecuadamente el esfuerzo a ejecutar.

El Modelo SQA orquesta y alinea las actividades definidas en las metodologías ARINbide y Probamet, estableciendo a su vez las actividades de aseguramiento de la calidad:



## Asignación del valor del NAC

El valor del NAC (Nivel de Aseguramiento de la Calidad) permite determinar el grado de calidad necesario para un proyecto de desarrollo software, y en consecuencia, identificar qué controles mínimos serán de obligado cumplimiento para el aseguramiento de la calidad del producto, así como las tipologías y niveles de pruebas a ejecutar.

Haciendo uso del formulario de cálculo “*Calculo\_NAC\_Proyecto*” el contratista **deberá** calcular el valor NAC (alto, medio, bajo o nulo) asociado al proyecto. No obstante, y puesto que es posible que algunos de los valores asociados a los criterios de cálculo todavía sean desconocidos, o bien porque el NAC obtenido no se considere adecuado para el proyecto dadas sus características, el contratista **puede** asignar un NAC distinto al calculado. En este caso, la modificación de NAC **deberá** estar totalmente justificada, especialmente en los casos en los que se asigne un NAC nulo.

## Plan SQA y actividades de aseguramiento de la calidad

Al inicio del proyecto, el Responsable del proyecto, de forma consensuada con la OTC constituida, **deberá** elaborar el Plan SQA del proyecto, estableciendo el detalle de las actividades de calidad a realizar, sus esfuerzos y fechas de compromiso según el Plan de Entregas obtenido en ARINbide y en función del NAC calculado.

El Modelo SQA contempla en general la realización de las siguientes actividades:

Proceso	Descripción
Verificación documental	Revisión de la documentación entregable de ARINbide y Probamet, realizada para asegurar que el formato y contenido de la documentación del proyecto cumple con requisitos necesarios
Auditoría de Código y Ejecución de Pruebas	El chequeo estático de código y las actividades de Pruebas Unitarias, de Integración y de Sistema, son tareas de aseguramiento de calidad del producto software definidas y pautadas en la metodología de pruebas Probamet
Auditorías de Fin de Fase SQA	Al finalizar cada una de las fases que marca ARINbide se asegura que se hayan generado todos los productos obligatorios, y que se hayan seguido apropiadamente las actividades establecidas
Ejecución selectiva de pruebas	Si se considera necesario, se contempla la ejecución selectiva de un conjunto de pruebas adicionales por parte de la OTC del proyecto con el objetivo de reforzar el proceso de Pruebas
Auditorías muestrales	Si se considera necesario, teniendo en cuenta la complejidad del proyecto y los niveles de calidad que se vayan obteniendo en las actividades SQA realizadas, se contempla la realización durante el proyecto de verificaciones de documentación y/o ejecución de pruebas adicionales, por parte de un equipo de calidad independiente (Oficina Técnica de Calidad de EJIE) que supervisa durante todo el proyecto las actividades de aseguramiento de calidad realizadas

Proceso	Descripción
<b>Auditoría Final de SQA</b>	Coincide con el paso previo a producción y consolida los resultados de las actividades de calidad realizadas en el proyecto y el grado de calidad obtenido en los productos generados. Establece conclusiones y recomendaciones para la toma de decisiones en el paso a producción del producto
<b>Recogida y Análisis de Indicadores (Cuadro de mando)</b>	Como resultado de las distintas actividades del modelo, se actualizan automáticamente los indicadores de calidad, ayudando también en la toma de acciones correctivas

## Verificación documental

La verificación de la documentación tiene por objetivo detectar cualquier no conformidad en una fase temprana del proyecto.

En base al Plan SQA, el contratista **deberá** asegurar la contratación de la ejecución de la verificación documental:

- Verificar que se genera la documentación necesaria y que contiene todo lo exigido por la metodología ARINbide. Para ello se **deberá** cumplimentar el formulario de “Checklist de Verificación de ARINbide”
- Verificar que se genera la documentación necesaria y que contiene todo lo exigido por la metodología Probamet. Para ello se **deberá** cumplimentar el formulario de “Checklist de Verificación de Probamet”

Las siguientes tablas muestran un resumen de las verificaciones documentales a realizar para cada metodología:

Fase de ARINbide	Verificación
<b>GPR – Gestión de Proyectos</b>	VPP – Verificación de la Planificación del proyecto (Plan de Entregas)
<b>ASI – Análisis de Sistemas de Información</b>	VCR – Verificación del Catálogo de Requisitos VDA – Verificación de la Documentación de Análisis
<b>DSI – Diseño de Sistemas de Información</b>	VDD – Verificación de la Documentación de Diseño
<b>CSI – Construcción del Sistema de Información</b>	VEC – Verificación de la Especificación de Construcción del Sistema VMU – Verificación Manual de Usuario
<b>IAS – Implantación de Sistema de Información</b>	VDI – Verificación Documentación de Implantación
<b>GCO – Gestión de la Configuración</b>	VGO – Validación de Gestión de Configuración

Fase de Probamet	Verificación
PPB - Planificación y Seguimiento de las Pruebas	VPB – Verificación Plan de Pruebas
APB - Análisis y Diseño de las Pruebas	VCP – Verificación Especificación de Casos de Prueba VMT – Verificación Matriz Trazabilidad Requisitos - Casos de Prueba

## Auditorías Fin de Fase SQA

Al finalizar todas las actividades previstas en el proyecto para cada fase de ARINbide, en base al Plan SQA del proyecto, la Oficina Técnica de Calidad del proyecto, **deberá** realizar las Auditorías de Fin de Fase planificadas, cuyos resultados y conclusiones se incluyen en el Informe de aseguramiento de la calidad ISQA.

El Modelo SQA contempla la realización de las siguientes:

1. Auditoría ASI: Coincide con el final de la fase de Análisis del Sistema
2. Auditoría DSI: Una vez terminada la fase de Diseño del Sistema
3. Auditoría CSI-I: Coincide con el final de las actividades de Construcción, Pruebas Unitarias y Pruebas de Integración
4. Auditoría CSI-II: Una vez terminadas las Pruebas Funcionales de Sistema
5. Auditoría CSI-III: Después de la realización de las Pruebas No Funcionales de Sistema: Pruebas de Prestaciones, Seguridad, Accesibilidad, Usabilidad y Pruebas de Fallo y Recuperación del Sistema

Para la realización de estas auditorías, se seguirán los correspondientes formularios de “Checklist de Auditorías SQA”, y de dispondrá de la información y documentación recopilada en cada fase, en base al cumplimiento del Modelo SQA.

## Auditoría Final SQA

Al finalizar todas las actividades de la fase de Implantación de ARINbide (IAS), y tras realizar las pruebas de Aceptación de Usuario, en base al Plan SQA del proyecto, la Oficina Técnica de Calidad del proyecto, **deberá** realizar la Auditoría Final SQA, cuyos resultados y conclusiones se incluyen igualmente en el Informe Final de aseguramiento de la calidad ISQA.

La Auditoría Final de SQA sigue la misma operativa que las Auditorías de Fin de Fase y obtiene de forma incremental el mismo informe pero dando la visión global del grado de calidad del proyecto y las conclusiones derivadas.



## Indicadores

El Modelo SQA define un Mapa de Indicadores que establece un conjunto unificado de métricas estándar para el desarrollo de proyectos y los umbrales de cumplimiento para cada uno de ellos. La recogida y análisis de Indicadores en los proyectos, se realiza con el soporte de un Cuadro de Mando que actualiza automáticamente los indicadores de calidad según se vayan realizando las tareas de SQA planificadas para el proyecto.

Se ha seleccionado un subconjunto básico de dichos indicadores que toda aplicación a implantar en el entorno de GV-EJIE **deberá** satisfacer, en función del NAC asignado al aplicativo. Se busca así asegurar con cierta garantía que los aplicativos implantados en el entorno productivo son lo suficientemente robustos y estables para su explotación.



Tras la realización de todas las actividades correspondientes según el modelo SQA en el entorno de Pruebas se obtendrán los indicadores para asegurar que el producto a instalar supera los umbrales establecidos para estos indicadores. La no superación de los límites establecidos podría suponer que el sistema no sea instalado en el entorno de producción.

El modelo SQA establece una serie de grupos de indicadores, cuyos valores umbrales están recogidos en el documento “Indicadores de calidad software”.

Existen indicadores de obligado cumplimiento que **deberán** contemplarse, con los umbrales estipulados en el mencionado documento, y que son los siguientes:

Indicadores	Descripción
de pruebas	Análisis estático de código Pruebas unitarias Pruebas de integración Pruebas de sistema: <ul style="list-style-type: none"> <li>• Pruebas basadas en requisitos (tanto funcionales como no funcionales)</li> <li>• Rendimiento (prestaciones): cumplimiento global indicadores prestaciones, indicadores servidor de aplicaciones, indicadores servidor web, indicadores servidor BBDD</li> </ul>
pruebas de sistema	Seguridad: grado cumplimiento owasp top 10 Usabilidad: grado cumplimiento evaluación heurística Accesibilidad
de calidad SQA	Indicadores de Fin de Fase

# Metodología de desarrollo ARINbide

## Alcance

ARINbide v1.0 se concibe como una metodología práctica de desarrollo para el ciclo de vida completo de los sistemas de información, basada en MÉTRICA Versión 3, para desarrollos orientados a objetos. Es la herramienta metodológica corporativa que sistematiza las actividades a realizar en los proyectos de desarrollo de aplicaciones y estandariza los entregables a obtener.

La propia metodología, como parte de sus procesos, actividades y tareas, permite generar de manera habitual todos los registros y evidencias necesarios para el sistema de aseguramiento de la calidad dictados por la norma ISO 9001:2000 implantado en el Gobierno Vasco.

En su proceso principal de ingeniería del software (ISW) describe y normaliza la secuencia de fases y actividades a realizar en el proyecto de elaboración de un sistema de información, así como los entregables a obtener en cada una de ellas. En este ámbito, ARINbide v1.0 contempla el desarrollo de las siguientes fases:

1. Catálogo de Requisitos de Usuario (CRU) (proceso diferenciado dentro del Análisis del Sistema de Información, ASI)
2. Análisis del sistema (ASI)
3. Diseño del sistema (DSI)
4. Construcción del sistema (CSI)
5. Implantación y aceptación del sistema (IAS)

El ciclo de vida descrito en la metodología de desarrollo ARINbide se alinea y complementa a lo largo del tiempo con la metodología de pruebas corporativa Probamet, centrándose ésta última en la especificación de todas las actividades relacionadas con la planificación, definición y ejecución de los diferentes tipos de pruebas. Existe una correspondencia entre las fases de ejecución de ARINbide y las fases de ejecución de Probamet.

ARINbide contempla además el establecimiento de otros procesos necesarios y que conviven en el tiempo con el proceso principal de ingeniería del software:

- Gestión de Proyectos (GPR): facilita la planificación, el seguimiento y control de las actividades y de los recursos humanos y materiales que intervienen en el desarrollo de un Sistema de Información, la Gestión de Riesgos de Proyecto
- Gestión de Riesgos de Proyectos (GRP): facilita la prevención y la minimización de riesgos para el proyecto, mediante la identificación y análisis de riesgos, la planificación de acciones, su registro y control
- Gestión de la Configuración (GCO): permite establecer el modo como se va a mantener la integridad y trazabilidad de la calidad de los productos software y documentales durante todo el ciclo de vida

- Mantenimiento de Sistemas de Información (MSI): proporciona el modelo de gestión de los servicios de mantenimiento de aplicativos una vez que están implantados, en base a un Acuerdo de Nivel de Servicio

Para los distintos procesos de ARINbide, ésta detalla la secuencia de fases, actividades y tareas a realizar, así como los entregables de cada una de ellas, y las técnicas sugeridas para su realización. El contratista **deberá** contemplar la contratación de la ejecución de dichas actividades definidas en base a los requisitos del proyecto.

## Ingeniería del Software (ISW)

El proceso principal de la metodología ARINbide define las siguientes fases:

### Catálogo de Requisitos de Usuario

Tomando como referencia la metodología Metrica V3, el Análisis del Sistema de Información (ASI) es la primera fase de un proyecto de desarrollo de software, una vez obtenidos los resultados de las fases de planificación y estudio de viabilidad de los sistemas, realizadas con anterioridad. En ARINbide se ha extraído la obtención de los requisitos de usuario como fase independiente del ASI para poner en valor uno de los entregables fundamentales del proceso de ingeniería del software, como es el Catálogo de Requisitos de Usuario o CRU.

El CRU contempla las siguientes actividades:

Actividad	Objetivo	Entregables
ASI 1. Definición del sistema	Describir el sistema y sus objetivos, delimitando su alcance, estableciendo las interfaces con otros sistemas e identificando a los usuarios representativos	
ASI 2. Establecimiento de requisitos	Definir, analizar y validar los requisitos a partir de la información facilitada por el usuario para obtener un catálogo detallado de requisitos, a partir del cual se pueda comprobar que los productos generados en las actividades de modelización se ajustan a los requisitos de usuario	Catálogo de requisitos
ASI 11.1. Presentación y aprobación del catálogo de requisitos	Realizar la presentación del catálogo de requisitos de usuario para su aprobación formal	Acta de reunión (registro de aprobación del CRU)

## Análisis del Sistema de Información (ASI)

Una vez aprobado el Catálogo de Requisitos de Usuario, define la secuencia de actividades y tareas que obtendrán como resultado final la Especificación de Requisitos del Sistema.

Actividad	Objetivo	Entregables
<b>ASI 3. Identificación de subsistemas de Análisis</b>	Facilitar el análisis del sistema de información llevando a cabo la división del sistema en subsistemas	
<b>ASI 4. Análisis de casos de uso</b>	Identificar las clases cuyos objetos son necesarios para realizar un caso de uso y describir su comportamiento mediante la interacción de dichos objetos	
<b>ASI 5. Análisis de clases</b>	Describir las clases identificadas para cada subsistema, especificando las responsabilidades que tienen asociadas, sus atributos y las relaciones entre ellas	
<b>ASI 8. Definición de interfaces de usuario</b>	Analizar los procesos del sistema en los que se requiere una interacción con el usuario con el fin de crear una interfaz que satisfaga todos los requisitos establecidos, teniendo en cuenta los perfiles a quienes va dirigido. Principalmente, estas interfaces son formatos de pantalla, diálogos e informes	
<b>ASI 9. Análisis de consistencia</b>	Garantizar la calidad de los distintos modelos generados en el proceso, asegurar que los usuarios y analistas tienen el mismo concepto del sistema y elaborar un documento de referencia para las siguientes fases del diseño y desarrollo	ERS – Especificación de requisitos del sistema
<b>ASI 10. Especificación del plan de pruebas</b>	Iniciar la definición del plan de pruebas especificando el marco general y estableciendo los requisitos de pruebas de aceptación (ver Probamet)	(ver Probamet)
<b>ASI 11.2. Presentación y aprobación del ERS</b>	Realizar la presentación del análisis del sistema de información para su aprobación formal	Acta de reunión (Registro de aprobación del ERS)

Las actividades ASI 3 (Identificación de Subsistemas de Análisis), ASI 4 (Análisis de casos de uso), ASI 5 (Análisis de Clases) y ASI 6 (Definición de Interfaces de usuario), se realizarán de forma paralela en el tiempo, y obtendrán distintos productos que conforman finalmente el documento de ERS.

La actividad ASI 10 (Especificación del plan de pruebas) se realiza como tal según la metodología de pruebas Probamet.

El documento ERS aprobado servirá tanto como información de entrada para las tareas de diseño, como de registro de verificación y validación de la etapa de análisis.

## Diseño del Sistema de Información (DSI)

El objetivo principal de esta fase es la entrega de la arquitectura a seguir en el sistema así como las especificaciones de construcción. También se realizará una revisión de los elementos definidos en la fase anterior, para conseguir un perfecto refinamiento y de esta manera tener más claro el sistema a desarrollar. Igualmente se provee la información requerida para la correcta implantación, operación y soporte del sistema.

Actividad	Objetivo	Entregables
<b>DSI 1. Definición de la arquitectura del sistema</b>	Describir las particiones físicas del sistema, la descomposición lógica en subsistemas de diseño y su ubicación en cada partición, así como la infraestructura tecnológica para dar soporte al sistema	
<b>DSI 3. Diseño de casos de uso reales</b>	Especificar el comportamiento de los casos de uso e interfaces de diseño	
<b>DSI 4. Diseño de clases</b>	Identificar las clases adicionales según el nuevo estudio de los escenarios de los casos de uso	
<b>DSI 6. Diseño físico de datos</b>	Diseñar el modelo físico de datos especificando los caminos de acceso a los datos persistentes del sistema	
<b>DSI 7. Verificación y aceptación de la arquitectura</b>	Garantizar la calidad de las especificaciones del diseño del sistema de información y la viabilidad del mismo, como paso previo a la generación de las especificaciones de construcción	EDS – Especificación de diseño del sistema
<b>DSI 8. Generación de especificaciones de construcción</b>	Generar las especificaciones para la construcción del sistema de información, a partir del diseño detallado	ECS – Especificación de construcción del sistema
<b>DSI 10. Especificación técnica del plan de pruebas</b>	Especificar el detalle del plan de pruebas del sistema de información por cada uno de los niveles de prueba (Ver Probamet)	(Ver Probamet)
<b>DSI 11. Establecimiento de requisitos de los procesos de soporte</b>	Completar los requisitos registrados en la arquitectura del sistema con aquellos relacionados con los procesos de soporte del sistema en el entorno de operación	Requisitos de los procesos de soporte (ver procesos y procedimientos vigentes)
<b>DSI 12. Aprobación del DSI</b>	Realizar la presentación del diseño del sistema de información para su aprobación formal	Acta de reunión (registro de aprobación del DSI)

Las actividades DSI 3 (Diseño de casos de uso reales), DSI 4 (Diseño de clases), DSI 6 (Diseño físico de datos) se realizarán de forma paralela en el tiempo.

La actividad DSI 10 (Especificación técnica del plan de pruebas) se realiza como tal según la metodología de pruebas Probamet.

## Construcción del Sistema de Información (CSI)

El objetivo principal de esta fase es la realización del producto software definido en la fase de diseño, así como la ejecución del plan de pruebas según describe Probamet.

Actividad	Objetivo	Entregables
<b>CSI 1. Preparación el entorno de desarrollo</b>	Asegurar la disponibilidad de todos los medios y facilidades para que se pueda llevar a cabo la construcción del sistema de información	
<b>CSI 2. Generación del código de los componentes</b>	Codificar los componentes del sistema de información, a partir de las especificaciones de construcción, así como la construcción de los procedimientos de operación y seguridad establecidos para el mismo	Producto software
<b>CSI 3-5. Ejecución de las pruebas</b>	Realizar las pruebas unitarias, de integración y de sistema (Ver Probamet)	(Ver Probamet)
<b>CSI 6. Elaboración del manual de usuario</b>	Obtener el manual de usuario	Manual de usuario
<b>CSI 9. Aprobación del sistema de información</b>	Realizar la presentación del sistema de información para su aprobación formal	Acta de reunión (registro de aprobación del sistema)

Las actividades CSI 3, 4 y 5 (Realizar las pruebas unitarias, de integración y de sistema) se llevan a cabo como tal según la metodología de pruebas Probamet.

## Implantación y Aceptación del Sistema (IAS)

El objetivo principal de este proceso es la entrega y aceptación del sistema así como el paso a producción del mismo.

Actividad	Objetivo	Entregables
<b>IAS 1. Elaboración del manual de implantación</b>	Elaborar el manual de implantación	Manual de implantación
<b>IAS 2. Elaboración del manual de explotación</b>	Elaborar el manual de explotación	Manual de explotación
<b>IAS 3. Traspaso al entorno de pruebas</b>	Incorporar el sistema al entorno de pruebas	
<b>IAS 5 – 6. Ejecución de las pruebas de aceptación</b>	Comprobar que se satisfacen los requisitos de usuario en las condiciones en las que se realizará la operación real	
<b>IAS 9. Presentación y aprobación del sistema</b>	Formalizar la aprobación del sistema mediante una presentación general al Comité de Dirección y la confirmación de su aprobación	Acta de reunión (registro de aprobación formal del sistema en producción)
<b>IAS 10. Paso a producción</b>	Establecer el punto de inicio en el que el sistema pasa a producción y se traspa la responsabilidad al equipo de mantenimiento	Sistema en producción

El paso a producción del sistema vendrá condicionado también a lo dispuesto en el Modelo de Aseguramiento de la Calidad (Modelo SQA), y al resultado de las pruebas realizadas.

# Metodología de pruebas Probamet

## Alcance

Probamet establece el modelo de actividades a seguir referente al proceso de pruebas de un producto software. Proporciona una visión global del proceso de pruebas del software que se construya, analizando en detalle cada una de las fases que forman el ciclo de pruebas, y describiendo, para cada una de ellas, las actividades a realizar y la documentación de entrada y salida que las conforman.

El proceso de pruebas definido en Probamet estará completamente alineado con ARINbide.

El contratista **deberá** contemplar la contratación de la ejecución de las actividades definidas por Probamet de forma paralela a las fases de desarrollo definidas en ARINbide, y como complemento necesario a ésta. El Nivel de Aseguramiento de la Calidad asignado al proyecto permitirá determinar los niveles y tipologías de pruebas que deben contemplarse, y por tanto que deberán aplicarse según la metodología.

## Metodología

Probamet describe todas las actividades de pruebas del producto software y se divide en las siguientes fases:

1. Planificación y Seguimiento de las Pruebas – PPB
  - a. Planificación de las Pruebas – PPB1
  - b. Seguimiento de las Pruebas – PPB2
2. Análisis y Diseño de las Pruebas – APB
3. Ejecución de las Pruebas – EPB

### Planificación de las Pruebas (PPB1)

El objetivo de la fase es determinar la viabilidad de las pruebas para cada una de las funcionalidades que componen el desarrollo, junto con el alcance de las mismas, qué tipos de pruebas se van a realizar así como la estimación de tiempo y recursos necesarios para llevarlas a cabo.

El principal resultado de esta fase es el Plan de Pruebas.

Se contemplan las siguientes actividades:



Actividad	Objetivo	Entregables
<b>PPB 1.1 – Definir el Alcance y la Estrategia de Pruebas</b>	Teniendo en cuenta tanto los requisitos funcionales como no funcionales del sistema, identificar los Niveles de Prueba y Tipos de Prueba a ejecutar, consideraciones en cada tipo de prueba, elementos que quedan fuera del alcance de las pruebas, identificación de las herramientas a utilizar, y definición del alcance de automatización	
<b>PPB1.2 – Definir el Entorno de Pruebas</b>	Describir el entorno general de pruebas y necesidades tanto software como hardware, y la infraestructura que se utilizará para realizar las pruebas, incluyendo los sistemas horizontales implicados	
<b>PPB1.3 – Determinar criterios de ejecución de las pruebas</b>	Indicar si es necesario las actividades previas a las pruebas, con el fin de asegurar su adecuado comienzo, y establecer los criterios para el inicio, suspensión y finalización de las mismas	
<b>PPB1.4 – Organizar el Equipo de Pruebas</b>	Establecer la organización y estructura del equipo de pruebas, indicando los perfiles que lo van a integrar, además de la posible participación de otros grupos de soporte, e identificar las necesidades de formación para los recursos identificados	
<b>PPB 1.5 – Estimar el Esfuerzo y Duración de Pruebas</b>	Partiendo de la estimación del proyecto y la documentación del análisis se realiza la estimación del esfuerzo y duración necesarios para poder abordar el alcance de las pruebas	
<b>PPB 1.6 – Identificar los Riesgos</b>	Identificar los principales riesgos que pueden afectar a la ejecución del Plan de Pruebas, y su impacto en el proyecto global, así como las acciones a tomar para su gestión y control	FRPB - Factores de Riesgo de Pruebas
<b>PPB 1.7 – Elaborar el Plan de Pruebas</b>	Con la información obtenida de las actividades anteriores se obtiene el Plan de Pruebas. Además se establece el procedimiento a llevar a cabo para la gestión de las incidencias detectadas.	PLPB - Plan de Pruebas PRPB - Planificación de pruebas (Project de Pruebas)
<b>PPB 1.8 – Aprobación del Plan de Pruebas</b>	Verificación y aprobación del Plan de Prueba, comprobando su consistencia contra los Requisitos establecidos para el proyecto y el Plan de Entregas	Acta de reunión

## Seguimiento de las Pruebas (PPB2)

El objetivo de la fase es realizar el seguimiento periódico de la actividad de pruebas con base en el Plan de Pruebas elaborado en la fase PPB1 de Planificación de Pruebas y si fuera necesario, actualizar dicho plan y toda la documentación asociada (planificación y factores de riesgo).

Es necesario realizar un seguimiento activo de la actividad de pruebas, por lo que es indispensable marcar su periodicidad al inicio del proyecto e indicarlo en el Plan de Pruebas.

Se contemplan las siguientes tareas:

- Realizar el seguimiento del Plan de Pruebas
- Revisión de la planificación y actualización del Project de pruebas
- Revisión de los riesgos actualizando el detalle de los riesgos ya definidos
- Incluir nuevos riesgos identificados
- Generación de Informes de Seguimiento de pruebas

Como resultado, se dan los siguientes entregables:

- Plan de Pruebas PLPB actualizado (si fuera necesario)
- Planificación de pruebas PRPB actualizado
- Documento actualizado de Factores de Riesgo de Pruebas FRPB
- Actas de Seguimiento (AR)
- Informes de Seguimiento de Pruebas ISPB

## Análisis y Diseño de las Pruebas (APB)

En base a la revisión detallada del sistema, los requisitos y el diseño detallado del mismo, se debe hacer la especificación de los casos de prueba, crear la matriz de trazabilidad que garantice una adecuada cobertura de los requisitos de negocio por los casos de prueba, y realizar todas las tareas de preparación previas a la ejecución de pruebas.

Se describen las siguientes actividades:

Actividad	Objetivo	Entregables
<b>APB 1 – Especificación de Casos de Prueba</b>	Especificar y priorizar los casos de prueba, creando la estructura e identificadores necesarios en la herramienta de gestión de pruebas, y definir los componentes necesarios para los casos de prueba automatizables	ECPB -Especificación de Casos de Prueba
<b>APB 2 – Matriz de Trazabilidad</b>	Crear la matriz de trazabilidad para mantener la asociación entre requisitos propios de pruebas y casos de pruebas, garantizando una adecuada cobertura de los requisitos de negocio.	MT - Matriz de Trazabilidad
<b>APB 3 – Tareas de preparación de Pruebas</b>	Realizar todas las tareas de preparación previas a la ejecución de pruebas y cumplimentar la información necesaria para preparar la ejecución de las mismas	ECPB - Especificación de Casos de Prueba actualizado PLPB - Plan de Pruebas actualizado

## Ejecución de las Pruebas (EPB)

En esta fase se prepara el entorno y las herramientas necesarias para la ejecución de las pruebas, y se ejecutan las pruebas, registrando los resultados en las herramientas de gestión definidas.

Asimismo se recogen y analizan todas las métricas establecidas para el proyecto y se generan los informes correspondientes recogiendo las conclusiones derivadas del análisis, la toma de decisiones y las mejoras propuestas.

La ejecución de las pruebas seguirá el siguiente ciclo:

- Análisis estático de código
- Pruebas Unitarias: probando cada módulo o componente por separado
- Pruebas de Integración: con el objetivo de probar los interfaces entre módulos e ir incrementando la prueba de los módulos y sistemas
- Pruebas de Sistema: sobre el sistema completo. El alcance de las pruebas de sistema y los tipos de pruebas, estará en función de los requisitos, distinguiéndose Pruebas del Sistema Funcionales y Pruebas del Sistema No Funcionales
- Pruebas de Aceptación: pruebas del usuario

En el Plan de Pruebas del proyecto se habrá determinado el alcance concreto, incluyendo los tipos de pruebas relevantes según los requisitos de la aplicación y su nivel NAC.

Se describen las siguientes actividades:

Actividad	Objetivo	Entregables
<b>EPB 1 – Configurar el entorno y las herramientas de ejecución</b>	Preparación del entorno y las herramientas de pruebas que se utilizarán para la ejecución y el almacenamiento de los resultados de las mismas, así como preparación de datos de prueba.	
<b>EPB 2 – Ejecución de las Pruebas</b>	Ejecutar las pruebas, realizando una adecuada gestión de incidencias y registrando los resultados en la herramienta de gestión de pruebas	Resultados de la ejecución de pruebas Resultados de las pruebas en la Herramienta de Gestión de Pruebas Registro de Incidencias en la Herramienta de Gestión de Incidencias
<b>EPB 3 – Generación y Análisis de Informes</b>	Generar los informes de seguimiento, en cada nivel de pruebas y el final de pruebas tras la ejecución de los casos de prueba, mostrando el estado del proyecto	ISPB - Informe de Seguimiento de las Pruebas INPB - Informe de Nivel de Pruebas IFPB - Informe Final de Pruebas

# Utilidades de Desarrollo de Aplicaciones (UDA)

## Introducción

UDA es un conjunto de herramientas, tecnologías, componentes y normativas funcionales y técnicas que permiten acelerar y normalizan el proceso de construcción de aplicaciones JEE en el ámbito de Gobierno Vasco – EJIE

- Determina la arquitectura conceptual básica de los nuevos sistemas JEE, así como la selección y configuración de tecnologías que implementan sus componentes
- Potencia la ayuda al desarrollo, posibilitando mediante herramientas la generación de forma automatizada y asistida del código de las nuevas aplicaciones a construir, en coherencia con la arquitectura definida
- Proporciona componentes reutilizables y adaptables que implementan patrones de interacción para la construcción de aplicaciones de Internet enriquecidas (RIA)

## Arquitectura Tecnológica

Uno de los principios básicos en la definición de toda arquitectura para posibilitar gran parte de los requisitos establecidos (Extensibilidad, mantenibilidad, facilidad de manejo) es sin duda alguna la división de responsabilidades o la creación de módulos con una finalidad en concreto, evitando mezclar diferentes aspectos de la creación. A su vez, la arquitectura tecnológica está basada en estándares.

Se han seleccionado una serie de tecnologías para utilizar en los diferentes módulos, que se detallan a continuación:

Ámbito	Tecnología
capa Presentación	utiliza componentes RIA: <ul style="list-style-type: none"> <li>• Cliente: JavaScript, CSS y AJAX                             <ul style="list-style-type: none"> <li>○ Vista: jQuery + jQuery UI → Componentes RIA [JSON]</li> </ul> </li> <li>• Servidor: JSP                             <ul style="list-style-type: none"> <li>○ Controlador: Spring MVC</li> </ul> </li> </ul>
Modelo	POJOS que definen entidades
Remoto	<ul style="list-style-type: none"> <li>• EJB 3.0 para la comunicación remota</li> <li>• Implementación mediante interfaces remotas</li> <li>• Wrappers para el despliegue</li> </ul>

Ámbito	Tecnología
Servicios de negocio	POJOS para la implementación Fachada para las peticiones <ul style="list-style-type: none"> <li>• Locales</li> <li>• Remotas</li> </ul>
Acceso a datos	POJOS para la implementación APIs de persistencia <ul style="list-style-type: none"> <li>• JDBC → SpringJDBC</li> <li>• JPA 2.0</li> </ul>
Librería de funciones comunes	(x38ShLibClasses-x.x.jar) Se integra en las aplicaciones a nivel de EAR Solventa requerimientos estructurales de EJIE Podrá evolucionar según se detecten nuevos requerimientos Bajo acoplamiento con las aplicaciones desarrolladas con UDA (AOP) Funcionalidades incluidas: <ul style="list-style-type: none"> <li>• Seguridad (incluye adaptador XLNetS)</li> <li>• Validación (facilita validación de campos en servidor)</li> <li>• Trazas (aplica formato estándar para Bitácora)</li> <li>• Utilidades y complementos</li> </ul>
Resolución de dependencias / librerías	Apache Maven <ul style="list-style-type: none"> <li>• Librerías a nivel de EAR [pom.xml]</li> </ul>

## Patrones de interacción

UDA ofrece una serie de componentes (RUP) que implementan un grupo de patrones de interacción de usuario identificados:

- «**Feedback**»: Canal informativo por el que se comunica al usuario los posibles errores o problemas acaecidos, los resultados de su interacción con los elementos de la aplicación y las posibles medidas a tomar para solucionar ciertas inconsistencias
- **Mensaje**: Tiene como objetivo mostrar al usuario de forma homogénea, clara y llamativa, los posibles mensajes que pueden desencadenar las acciones en la aplicación. Estos mensajes predefinidos pueden ser de diferente tipología: error, confirmación, aviso o alerta
- **Diálogo**: Permite lanzar un subproceso o un mensaje de confirmación dentro de un proceso principal sin salirse de este. Es una evolución del patrón mensaje
- **Idioma**: El componente de idioma está diseñado para permitir al usuario elegir de forma intuitiva el idioma en el que se le presenta la aplicación

- **Menú:** Menú de la aplicación mantenido a lo largo de todas las páginas de forma consistente que muestra entradas directas a secciones clave de la aplicación
- **“Migas de pan”:** Muestra a los usuarios los pasos de navegación seguidos durante su interacción con la aplicación y les facilita enlaces para volver a páginas precedentes o la página de inicio
- **«Combo»:** Permite al usuario recuperar un elemento de una gran lista de elementos o de varias listas dependientes de forma sencilla y ocupando poco espacio en la interfaz
- **Autocompletar:** En cuanto el usuario comienza a escribir una búsqueda se le sugieren búsquedas relacionadas con lo que ha escrito que pueden ser de su interés
- **Fecha:** Permite al usuario introducir y seleccionar una fecha, tanto de forma manual como visual, moviéndose fácilmente por días, meses y años. Además, para minimizar las posibilidades de introducir una fecha incorrecta, ofrece al usuario ayudas y sugerencias de formato
- **Hora:** El usuario puede introducir y seleccionar una hora tanto de forma manual como visual, moviéndose fácilmente por las horas y los minutos, recibiendo ayudas y sugerencias para minimizar las posibilidades de introducir una hora incorrecta
- **Botonera:** Se les presenta a los usuarios una barra de botones con diversas funcionalidades relacionadas con elementos de la página. Gracias a este componente se presentan, ordenan y agrupan las distintas funcionalidades gestionadas por las aplicaciones
- **Tabla:** Se les presenta a los usuarios los datos tabulados para que la información se visualice de manera ágil y rápida, facilitando así su comprensión y manejo
- **Mantenimiento:** Implementa un nuevo patrón definido para facilitar la lógica necesaria en las acciones básicas, denominadas CRUD (create, read, update y delete), sobre una tabla
- **Pestañas:** Permiten dar acceso de forma compacta a grupos de contenidos mutuamente excluyentes pudiendo ser integradas en zonas muy reducidas de la interfaz
- **«Tooltip»:** Todas las aplicaciones deben tener un sistema de ayuda aunque algunas necesiten uno más completo y preciso que otras. El objetivo del sistema de ayuda debe ser, por un lado ayudar al impaciente y/o al usuario ocasional tan extensamente como sea posible y, por otro, ayudar a los usuarios expertos

## Generadores de código

UDA posee asistentes para generación de proyectos, código y de vista, utilizando plugins del IDE Eclipse:

- Asistentes para generación de proyectos
- Asistentes para generación de código (en base al modelo relacional)
- Asistentes para generación de vista (interfaces de usuario)

## Kit de desarrollo

UDA se presenta como un Kit de desarrollo, el cual contiene:

- Eclipse Helios OEPE
- Apache Maven (resolución de dependencias / librerías)
- Plugin UDA:
  - Plugin Eclipse
  - Plantillas (templates)
- Reglas de validación de la calidad del código:
  - PMD / Checkstyle / FindBugs

Para todo esto se requiere como software de base:

- Servidor de aplicaciones Weblogic 11g
- Java 6
- Acceso a Base de Datos Oracle 10g
- Repositorio SVN