Análisis y Diseño

Playbook para desarrolladores de software

Transformación desarrollo de *software* - #ONE

Control de versiones

Versión	Fecha de creación	Responsable	Descripción
1.0	{24-05-2023}	GSD (Global Software Development)	Primera versión oficial de los playbooks

Índice

1. Introducción **4**
1.1. Acerca de este Playbook **4**
1.2. Principios básicos **5**
1.3. Niveles de exigencia **5**
2. Prácticas **6**
2.1. Racionalizar, minimizar y unificar entregables **7**
2.1.1 Racionalizar, minimizar y unificar entregables 7
2.2. Asegurar que los equipos sean autosuficientes **14**
2.2.1 Asegurar que los equipos sean autosuficientes 14
2.3. Aplicar un diseño orientado a servicios **18**
2.3.1 Aplicar un diseño orientado a servicios 18
2.4. Reutilizar, desacoplar y modularizar componentes **20**
2.4.1 Reutilizar, desacoplar y modularizar componentes 20
2.5. Garantizar que la seguridad está implícita en el diseño de soluciones **23**
2.5.1 Garantizar que la seguridad está implícita en el diseño de soluciones 23
3. Proceso **26**
Anexo **35**
Criterio de análisis de Seguridad (anteriormente llamado modelo Not Required) 35
Ejemplo de jerarquía para el servicio Web Channel 36
Ejemplos de entregables 37

1. Introducción

1.1. Acerca de este Playbook

El presente playbook (parte de una serie de seis) es un elemento fundamental de las prácticas de desarrollo de software de el Banco. En él se definen los principios, el proceso, las prácticas y las herramientas que todos los equipos de desarrollo de software de el Banco deben adoptar durante el proceso de análisis y diseño. Este abarca la fase de análisis de una iniciativa de desarrollo de software, que se inicia una vez se aprueba su presupuesto hasta la validación de su modelo de solución (quedando así fuera del alcance de este Playbook las actividades realizadas en las etapas anteriores de Detección de Oportunidades y Review de la fase de ideación y en la fase posterior de ejecución de la iniciativa).

El playbook debe ser actualizado regularmente basándose en las prácticas y el panorama tecnológico de el Banco. El equipo de GSD (Global Software Development) será el responsable de actualizarlo.

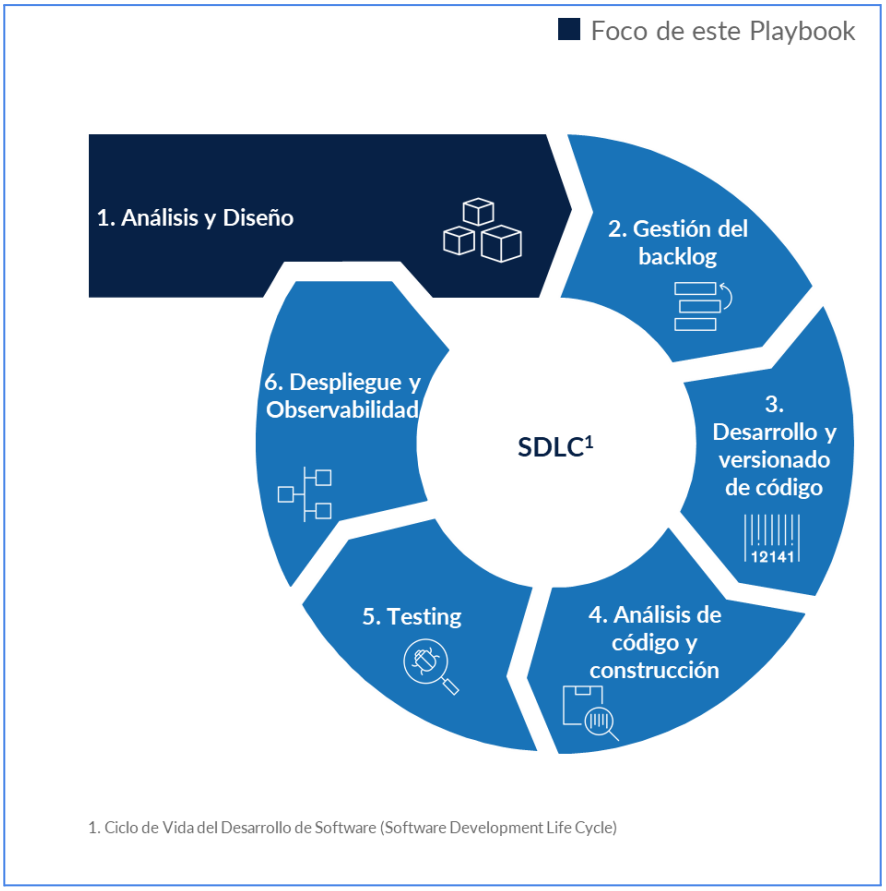


Figura 1 - Ciclo de vida del desarrollo de *software*

1.2. Principios básicos

Los seis playbooks se guían por un conjunto compartido de principios básicos:

- **Estandarización:** Usar una taxonomía global y formas de trabajo alineadas entre los equipos y geografías permitirá a los desarrolladores colaborar mejor, compartir mejor el conocimiento y rotar entre los equipos.
- **Calidad sobre cantidad:** Escribir código de calidad debe ser nuestra prioridad ya que la mayor parte de nuestro tiempo lo empleamos manteniendo software heredado y abordando deuda técnica, cumplir con esto mejorará la velocidad de desarrollo de los equipos a medio y largo plazo.
- **Transparencia y trazabilidad:** El total control y visibilidad de los sistemas y el código de el Banco mejora la experiencia de los desarrolladores para navegar en los distintos entornos, responder a incidentes y garantizar el cumplimiento de los requisitos regulatorios.
- **Propiedad:** La alta demanda de calidad requerida en los servicios críticos del banco implica un control detallado de responsabilidad y propiedad de los activos de código. Debemos ser propietarios del código que desarrollamos y sentirnos orgullosos de ello.
- **Simplicidad:** Los sistemas de software rara vez se escriben una vez y no se modifican. Más bien, muchas personas trabajan con el mismo sistema a lo largo del tiempo. Por eso es importante buscar las soluciones más sencillas que puedan comprenderse fácilmente y evolucionar en el futuro.

1.3. Niveles de exigencia

Este playbook utiliza intencionadamente las siguientes tres palabras para indicar los niveles de exigencia según la norma **RFC2119**:

- **Debe** - Significa que la práctica es un requisito absoluto, salvo las excepciones que se autoricen de forma específica.
- **Debería** - Significa que pueden existir razones válidas en circunstancias particulares para ignorar una práctica, pero deben comprenderse todas las implicaciones y sopesar cuidadosamente antes de elegir un camino diferente.
- **Podría** - Significa que una práctica es realmente opcional.

2. Prácticas

Todos los equipos de *software* (desarrollo, arquitectura, seguridad,...) **deben** llevar a cabo las siguientes **cinco prácticas** durante el proceso de análisis y diseño:

2.1. Racionalizar, minimizar y unificar entregables
2.2. Asegurar que los equipos sean autosuficientes
2.3. Aplicar un diseño orientado a servicios
2.4. Reutilizar, desacoplar y modularizar componentes
2.5. Garantizar que la seguridad está implícita en el diseño de soluciones

Cada una de las prácticas se define más adelante con la siguiente **estructura**:

- Sus beneficios (qué conseguimos).
- Precondiciones (condiciones para poder aplicar la práctica).
- Adopción (cómo implementarla).
- Herramientas necesarias.
- Indicadores para los desarrolladores (cómo la medimos).
- Enlaces de interés.

2.1. Racionalizar, minimizar y unificar entregables

Beneficios

- Evitar duplicidad y retrabajo en los entregables del proceso de análisis y diseño.
- Agilizar el proceso de análisis y diseño sin penalizar la calidad de los entregables y la seguridad de los aplicativos.
- Facilitar la colaboración entre equipos de desarrollo.

Precondiciones

- Criterios definidos por el equipo de arquitectura que determinen si una iniciativa requiere un análisis de arquitectura.
- Criterios definidos por el equipo de seguridad que determinen si una iniciativa requiere un análisis de seguridad.
- Formación a los equipos sobre el proceso de análisis y diseño.
- Adopción de una taxonomía estándar de los servicios y entidades funcionales del banco (p.ej., pagos, préstamos, canales).
- Disponibilidad de la descripción técnica de alto nivel (e.g., diseños de solución, ficha de la iniciativa, Modelo de Seguridad) de los distintos servicios y componentes de software en la herramienta de documentación global. Estos deben ser accesibles por todos los equipos siempre que no sean componentes críticos.

Adopción

2.1.1. Racionalizar, minimizar y unificar entregables

El análisis de una nueva iniciativa[1] de software es el proceso de elaboración de la 1) definición funcional, 2) diseño y definición de la arquitectura a alto nivel y 3) definición del modelo de seguridad de la solución. Durante esta fase se generan hasta cuatro entregables para un servicio que **deben** ser racionalizados, minimizados y unificados para que otro equipo pueda conocer los aplicativos de ese servicio y la lógica técnica y funcional de los mismos. Estos entregables son:

- **Formulario de Categorización** - cuestionario conformado por tres bloques con un tiempo recomendado para cumplimentarlo de 25 mins:
    - **Criterio de análisis** – Breve formulario obligatorio que determina si **debe** realizarse un análisis de arquitectura y de seguridad en base a las características de la iniciativa y el criterio definido por los equipos de arquitectura y seguridad.
    - **Formulario de Admisión de Riesgos IT (FARIT)** – Formulario de riesgo tecnológico obligatorio que determina el nivel de riesgo de la iniciativa en las dimensiones de

[1] Una **iniciativa** es un nuevo proyecto de software, nuevas épicas de un proyecto o cambios en las épicas existentes de un proyecto que serán añadidos al backlog de tareas del proyecto. Para más información sobre la gestión del backlog visitar el playbook de Gestión del backlog.

Technology Security, Information & Data Security y Digital Fraud. El resultado del FARIT será uno de los factores utilizados por el Decision Framework para determinar el modelo de participación del Solution Architect y Seguridad en Proyectos en el análisis de la iniciativa.
    - **Decision Framework** – Formulario que determina el modelo de participación del Solution Architect o Seguridad en Proyectos en el análisis de la iniciativa. Este formulario sólo **debe** cumplimentarse si el criterio de análisis determina que es necesario un análisis de arquitectura o de seguridad. Para más información sobre los modelos de participación, ver la Práctica 2.2.
- **Ficha de la iniciativa** – Documento que contiene la descripción de la nueva iniciativa. Los puntos a incluir en la Ficha de la Iniciativa corresponden a los niveles 1, 2 y 3 del Documento de Análisis realizado hasta la fecha y deberán acordarse entre el Tech Lead, Solution Architect y Seguridad en Proyectos.
- **Diseño de Solución** – Documento que recoge a alto nivel la solución funcional y técnica de una iniciativa.
- **Modelo de Seguridad** – Documento que recoge el conjunto de controles y medidas de seguridad que deben aplicar al servicio o aplicativo.

Con el fin de fomentar la eficiencia y reducir la carga burocrática, **estos entregables deben generarse solo cuando sean necesarios y aportando un nivel de detalle acorde a la complejidad de la iniciativa** tal y como se describe a continuación.

El Tech Lead **debe** rellenar el **Formulario de Categorización** para todas las nuevas iniciativas, apoyándose en los perfiles especialistas que considere necesarios**.** Sin embargo, solo son obligatorios los dos primeros bloques: criterio de análisis y FARIT. El bloque Decision Framework solo **debe** cumplimentarse si el criterio de análisis determina que la iniciativa requiere un análisis.

El Tech Lead **debe** completar la **Ficha de la Iniciativa** solo si el bloque Criterio de Análisis del Formulario de Categorización determina que la iniciativa requiere un análisis. En este documento se introduce automáticamente el enlace a la entrada de la iniciativa en la SDA tool donde se describen sus requerimientos funcionales, beneficios y se indica si es una iniciativa global. Si el Decision Framework del Formulario de Categorización determina que un Solution Architect o Seguridad en Proyectos deben participar en la iniciativa, el Tech Lead debe aportar una descripción de la misma que permita la participación de los especialistas. El nivel de detalle requerido **debe** acordarse entre el Tech Lead, el Solution Architect y Seguridad en Proyectos acorde a la complejidad de la iniciativa. A continuación, se sugiere la información que el Tech Lead **debería** aportar en la Ficha de la Iniciativa:

- Geografías afectadas por el proyecto
- Casos de uso (features funcionales o entregables) y para cada uno:
    - Descripción

        - Entidades de datos que intervienen
        - Si requiere la ingesta de nuevos datos para el caso de uso o modificaciones de datos ya ingeridos
        - Si genera nuevos datos
        - Qué se necesita para consultar y/o modificar
        - Si requiere datos de recursos públicos o privados
        - Si existen datos críticamente confidenciales
        - Dónde se almacenan los datos requeridos
        - Si existe salida de información fuera de CPD
        - Dónde se van a consumir los datos
        - Tipo de consumo de los datos
    - Si existen requisitos legales o normativos a cubrir o impactados. El equipo de la iniciativa **debe** contactar al área jurídica o legal y describir los casos de uso de la iniciativa. La persona de contacto del área jurídica o legal **debe** definir si existen requisitos legales o normativos a cubrir o impactados
    - Si es necesario un backup fuera del estándar
    - Nivel de disponibilidad requerida (8x5 o 24x7)
    - Si es crítico para negocio. Para ello el Tech Lead debe contactar al equipo de Continuidad de Negocio de su geografía que le debe indicar la criticidad de los aplicativos afectados por la iniciativa.
    - Para cada caso de uso:
        - Canales y actores / roles que intervienen
        - Integración entre los sistemas intervinientes
        - Comunicaciones
        - Documentación y firma
    - Prototipo
    - Diagrama del proceso (si ha intervenido BPE): ARIS
    - KPIs funcionales y de negocio
    - Requisitos no funcionales
    - Documentación de seguridad aplicable o justificación de no aplicación

Debe crearse un **Diseño de Solución** solo si la sección criterio de análisis del Formulario de Categorización determina que la iniciativa requiere un análisis de arquitectura. **En caso de que se determine que no se requiere análisis de arquitectura, no habrá Diseño de Solución.**

Debe crearse un **Modelo de Seguridad** solo si la sección criterio de análisis del Formulario de Categorización determina que la iniciativa requiere un análisis de seguridad. **En caso de que se**

determine que no se requiere análisis de seguridad (lo que hasta ahora se conocía como modelo Not Required)**, no habrá entregable de seguridad ni se incluirán medidas o controles de seguridad en el Diseño de Solución.**
Los equipos **deben** evitar duplicar esfuerzos y documentación. Por lo tanto siempre que se necesite un nuevo análisis **deben** conocer el diseño vigente (si existiera) para un aplicativo y servicio y **realizar el análisis sobre este diseño, generando una nueva versión que evolucione la documentación existente** y evite la existencia de duplicados y versiones obsoletas que aparezcan activas en el repositorio de documentación.

Con este fin, los equipos **deben** disponer de una **herramienta de documentación global** (pero adaptable a las geografías) donde almacenar las Fichas de las Iniciativas, Diseños de Solución y Modelos de Seguridad de los servicios y aplicativos.

Dicha documentación **debe** tener un punto de entrada único y contar con dos espacios: uno para el servicio y otro para las iniciativas.

El espacio de servicio **debe** cumplir las siguientes directrices:

- El espacio de servicio **debe** contener tres páginas:
    - Una que almacene el **Diseño de Solución** del servicio
    - Una que almacene el **Modelo de Seguridad** del servicio
    - Una página dedicada a las **unidades aplicativas** que componen el servicio. En aquellos servicios en los que, por su complejidad, estos puedan facilitar su comprensión, esta página **debería** contener una página para cada unidad aplicativa del servicio y en cada una, el **Diseño de Solución y Modelo de Seguridad de la unidad aplicativa**.
- **Cada nueva versión** del diseño de solución del servicio o de un aplicativo **debe enlazarse con la Ficha de la Iniciativa** que provocó su evolución, que se encuentra en el espacio de iniciativas.
- El Diseño de Solución y el Modelo de Seguridad de un servicio **debe** enlazarse con los diseños de solución y los Modelos de Seguridad de las unidades aplicativas que lo componen, respectivamente.
- El Service Owner **debe** asegurar que la documentación de su servicio esté actualizada. Para más información sobre las responsabilidades del Service Owner ver el Playbook del Service Owner.
- El espacio de servicio **puede** evolucionar para incorporar páginas dedicadas a otras dimensiones aparte de seguridad y arquitectura (p.ej. Infraestructura, UX)

El espacio de iniciativas **debe** cumplir las siguientes directrices:

- El espacio de iniciativas **debe** contener una página para cada iniciativa para la que se ha registrado un issue tipo análisis en la herramienta de gestión del backlog y en cada página almacenar la Ficha de la Iniciativa correspondiente.
- Cuando la arquitectura de un servicio se vea afectada por una iniciativa (i.e. requiera cambios en su diseño de solución) se creará una nueva versión del Diseño de Solución. Cada Ficha de la Iniciativa **debe** enlazarse con esa nueva versión del Diseño de Solución y Modelo de Seguridad de los servicios impactados.

La Figura 2 muestra la jerarquía documental a seguir. Ver Anexo para un ejemplo ilustrativo de jerarquía para el servicio Web Channel.

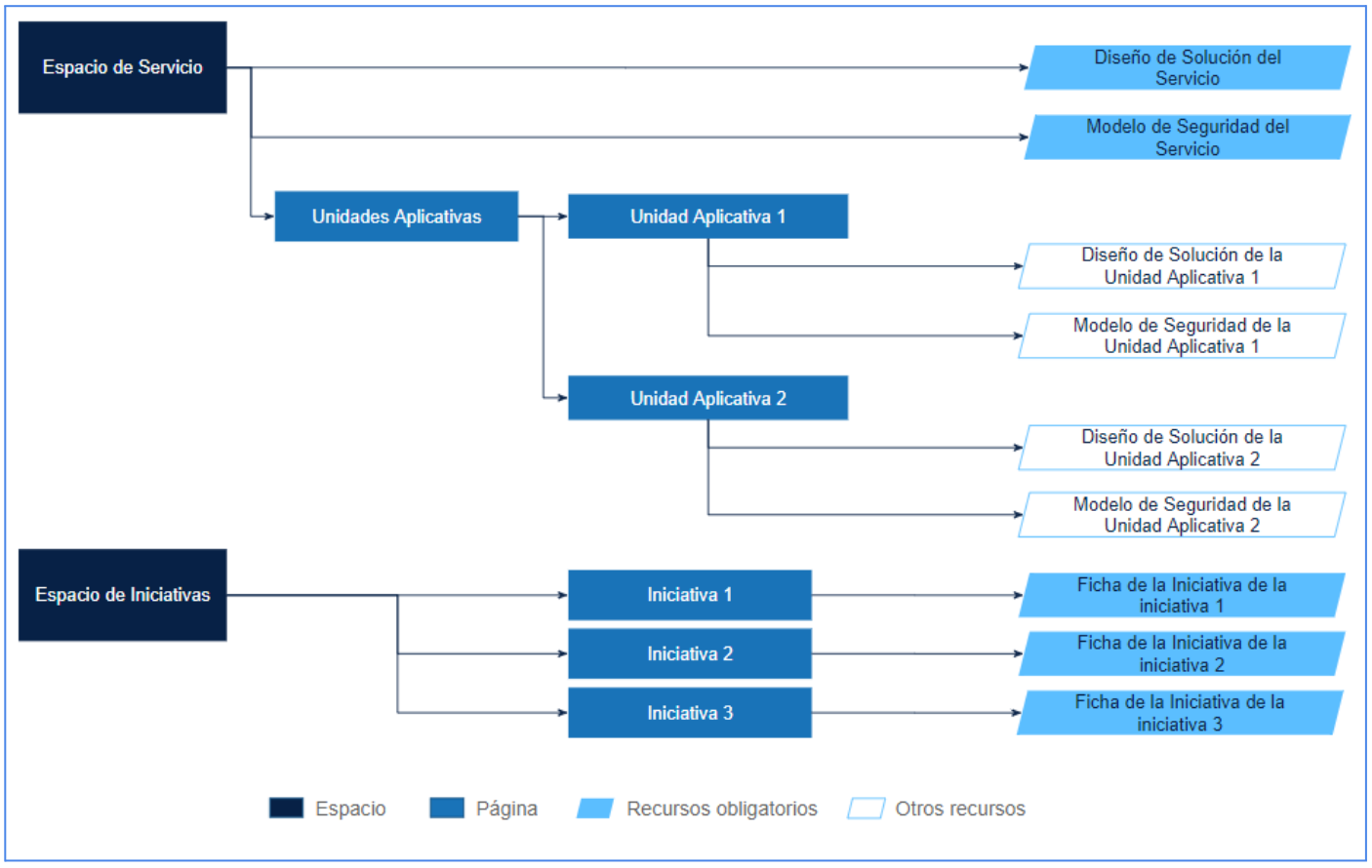


Figura 2 - Jerarquía de recursos en la herramienta de documentación global

En relación a esta práctica, los equipos de desarrollo **deberían** conocer los principios de agile architecture relacionados con la generación de entregables:

- **Colaboración evolutiva frente a entregables fijos** (blueprints)**:** Los equipos de desarrollo trabajan de forma conjunta permitiendo que los diseños evolucionen a lo largo del tiempo.
- **Comunicación frente a perfección:** En lugar de buscar la perfección en la documentación y el diseño detallado, los equipos de desarrollo deben centrarse en la comunicación efectiva entre los miembros del equipo y los interesados en el proyecto.
- **Modelos de alto nivel** (cuanto más complejos, más abstractos): Los modelos arquitectónicos deben ser de alto nivel, proporcionando una visión general del sistema en lugar de detalles específicos. Además, los modelos deben ser más abstractos cuanto más complejo sea el sistema.

Herramientas

La adopción de esta práctica requiere el uso de las siguientes herramientas:

- **Herramienta de documentación global:** Herramienta pendiente de definición.
- **Formulario de Categorización**: Formulario online que determine de forma automática si se requiere un análisis de arquitectura y de seguridad en base a la descripción de la iniciativa proporcionada por el Tech Lead y los criterios definidos por los equipos de arquitectura y seguridad.
- **Plantillas de entregables:** Para la Ficha de la Iniciativa, Diseño de Solución y Modelo de Seguridad

Indicadores

Los indicadores que apliquen deben ser proporcionados a los equipos para que puedan conocer su grado de madurez respecto a la práctica:

Indicador	Descripción
Porcentaje de iniciativas que no requieren análisis	Porcentaje de iniciativas para las que el Criterio de Análisis del Formulario de Categorización ha determinado que no requieren análisis sobre el total de iniciativas para las que se ha rellenado el Formulario de Categorización
Porcentaje de análisis almacenados en la herramienta de documentación global	Porcentaje de análisis almacenados en la herramienta de documentación global sobre el total de análisis realizados
Porcentaje de diseños de solución de servicio	Porcentaje de espacios de servicio en la herramienta de documentación global que almacenan un Diseño de Solución de servicio

Porcentaje de Modelos de Seguridad de servicio	Porcentaje de espacios de servicio en la herramienta de documentación global que almacenan un Modelo de Seguridad del servicio

Enlaces

- https://es.wikipedia.org/wiki/Desarrollo_%C3%A1gil_de_software

2.2. Asegurar que los equipos sean autosuficientes

Beneficios

- Reducir el tiempo requerido para diseñar la arquitectura de la iniciativa gracias a la minimización de las dependencias entre áreas.
- Mejorar la calidad de los diseños de arquitectura mediante una colaboración más efectiva.

Precondiciones

- Criterios que identifiquen el modelo de participación de Solution Architect en el proceso de análisis.
- Criterios que identifiquen el modelo de participación de Seguridad en Proyectos en el proceso de análisis.
- Descripción clara por parte del Project Sponsor de los beneficios, objetivos y requerimientos funcionales de la iniciativa.
- Facilidad para integrar perfiles expertos (p.ej. arquitectura, seguridad) cuando se inicie el análisis de las iniciativas.
- Formación a los equipos sobre el proceso de análisis y diseño.
- Formación del Tech Lead en materia de arquitectura y seguridad, diseñada por los equipos de arquitectura y seguridad

Adopción

2.2.1. Asegurar que los equipos sean autosuficientes

Los equipos de desarrollo de software **deberían** contar con la experiencia y habilidades necesarias para llevar a cabo de forma autónoma y ágil el proceso de diseño de arquitectura y seguridad de nuevas iniciativas (descrito en el capítulo de Proceso). Con este fin, el Tech Lead **debería** realizar de forma periódica **formaciones** diseñadas por los equipos de arquitectura y seguridad, desarrollando así sus capacidades en estas áreas. A pesar de ello, en aquellos casos en los que el Tech Lead no cuente con la experiencia y habilidades necesarias para realizar el diseño, **debe** integrarse de forma activa los perfiles de arquitectura y seguridad (Solution Architects y Seguridad en Proyectos) que se requieran en dichos equipos.

El Tech Lead **debe** completar el Decision Framework del Formulario de Categorización y puntualmente **puede** apoyarse en los perfiles especialistas que considere necesarios. Este formulario determinará el modelo de participación del Solution Architect y Seguridad en Proyectos en la iniciativa en función de las características de la misma y los criterios definidos por el equipo de arquitectura y seguridad.

Existen dos modelos de participación del Solution Architect para la creación del Diseño de Solución de una iniciativa:

- **Go systems:** El Solution Architect no participa en el diseño de la arquitectura de la iniciativa, que es realizado de forma autónoma por el Tech Lead. Este modelo absorbe la mayor parte de las iniciativas que se realizan bajo el modelo Lean hasta la fecha. Desde que se completa la Ficha de la Iniciativa hasta que todas las partes implicadas completan el diseño no **deberían** transcurrir más de 14 días naturales.
- **Specialist**: El diseño es realizado conjuntamente entre el Tech Lead y un Solutions Architect, que se integra en el equipo de la iniciativa. Desde que se completa la Ficha de la Iniciativa hasta que todas las partes implicadas completan el diseño no **deberían** transcurrir más de 50 días naturales.

Existen dos modelos de participación de Seguridad en Proyectos para la creación del diseño de seguridad de una iniciativa:

- **Security Review:** Seguridad en Proyectos establece las líneas generales de seguridad para la arquitectura de la iniciativa, el Tech Lead realiza el diseño de seguridad y este es revisado por Seguridad en Proyectos. Siempre y cuando el Tech Lead avise a Seguridad en Proyectos 15 días antes de la finalización del diseño de arquitectura, Seguridad en Proyectos **debería** completar la revisión en las dos semanas siguientes a la finalización del diseño de arquitectura.
- **Specialist:** El diseño de seguridad es realizado por Seguridad en Proyectos, que se integra en el equipo de la iniciativa, alineándolo con el Tech Lead para cumplir las necesidades funcionales de la iniciativa. Desde que se completa la Ficha de la Iniciativa hasta que todas las partes implicadas completan el diseño no **deberían** transcurrir más de 50 días naturales.

Además, en aquellas iniciativas en las que por su complejidad o las piezas que intervienen el Tech Lead, Solution Architect o Seguridad en Proyectos identifiquen que requieren experiencia y habilidades complementarias a las suyas para realizar el análisis de la iniciativa (p.ej., en materia legal o en el diseño de infraestructura on premise), estos **pueden** apoyarse en los perfiles especialistas necesarios. Cuando esto ocurra, el Tech Lead, Solution Architect o Seguridad en Proyectos:

- **Deben** identificar a los especialistas de los que se va a requerir una participación elevada durante la fase de análisis.
- **Deberían** conseguir un compromiso por parte de los especialistas para dedicar el tiempo suficiente a la realización del análisis.
- **Deben** establecer sesiones conjuntas periódicas para la realización del análisis y las responsabilidades de cada uno de los participantes.

Herramientas

Herramienta de gestión del backlog (Jira, donde pueda utilizarse)
Formulario de Categorización: (Criterio de análisis, Formulario de Admisión de Riesgos IT y Decision Framework). Formulario online que determine de forma automática el modelo de participación de Solution Architect y Seguridad en Proyectos en el análisis y diseño de una iniciativa en base a la descripción de la misma proporcionada por el Tech Lead y los criterios definidos por los equipos de arquitectura y seguridad.

Indicadores

Los indicadores que apliquen deben ser proporcionados a los equipos para que puedan conocer su grado de madurez respecto a la práctica:

Indicador	Descripción
Porcentaje de iniciativas bajo el modelo Go Systems y Specialist	Porcentaje de iniciativas abiertas bajo los modelos de participación del Solution Architect Go Systems y Specialist
Tiempo de entrega	Tiempo transcurrido desde que se completa la Ficha de la Iniciativa hasta que se valida el análisis
Porcentaje de iniciativas que cumplen los tiempos de entrega definidos	Porcentaje de iniciativas que cumplen el tiempo de entrega definido del modelo de participación asignado a la iniciativa
Tiempo de sincronización	Tiempo en el que la iniciativa permanece en espera, en estado “Waiting for Info”, al no recibir la información requerida para poder iniciar el análisis
Tiempo de atención por parte de shapers	Tiempo en el que la iniciativa permanece en espera debido a que en su análisis deben participar perfiles especialistas (p.ej. Solution Architect, Infraestructura, Seguridad en Proyectos) pero no disponen de capacidad
Tiempo de revisión del diseño	Tiempo dedicado a validar los entregables de Solution Architect y Seguridad en Proyectos, si los hubiera, previo al cierre final del análisis
Tiempo de bloqueo	Tiempo que la iniciativa no ha podido progresar debido a dependencias con otras áreas del banco

Enlaces

N/A

2.3. Aplicar un diseño orientado a servicios

Beneficios

- Mejorar la comunicación entre los equipos de desarrollo y negocio para evitar errores en el proceso de desarrollo y mejorar la calidad del diseño.
- Definir un diseño claro y unificado del aplicativo y por tanto mejorar el proceso de desarrollo posterior.
- Integrar el eje servicio en las arquitecturas, permitiendo unificar diseños y evitar redundancias.

Precondiciones

- Adopción de una taxonomía estándar de los servicios y entidades funcionales del banco (p.ej., pagos, préstamos, canales).

Adopción

2.3.1. Aplicar un diseño orientado a servicios

El Tech Lead **debe** interactuar con el negocio durante el análisis y diseño de soluciones de software y asegurarse de que los diseños cumplen con los requerimientos de negocio.

Los diseños de solución que se generen **deben** construirse en torno a los servicios existentes impactados por la iniciativa y aplicar las convenciones de nombres existentes (p.ej., definir las conexiones con un servicio de pagos usando la nomenclatura existente para ese servicio). Para este propósito, el Tech Lead o el Solution Architect se **pueden** apoyar en los Product Owners o Service Owners a lo largo del proceso de análisis y diseño.

Los equipos de desarrollo **deberían** aplicar el principio de Diseño Dirigido por Dominios (DDD). Este principio refleja la importancia de **modelar y entender el dominio funcional de negocio** - compuesto de entidades, atributos y relaciones relevantes para el negocio - en el que la aplicación opera, así como los procesos en los que éstas intervienen y son objeto de definición / modificación en la iniciativa. En el caso de el Banco, el dominio de negocio hace referencia al **servicio** implicado en la iniciativa. La idea principal es que los dominios de negocio suelen ser complejos y esa complejidad tiene que ser gestionada desde la fase de diseño de las soluciones para desarrollar soluciones efectivas, comprensibles y mantenibles. Por ello, los diseños de solución de alto nivel **deberían** reflejar las entidades funcionales de negocio (p.ej., representar dominio de cliente y pagos en una aplicación de transferencias).

Herramientas

La adopción de esta práctica requiere el uso de las siguientes herramientas:

- **Herramienta de documentación global:** Herramienta pendiente de definición.

Indicadores

N/A

Enlaces

- https://martinfowler.com/tags/domain%20driven%20design.html

2.4. Reutilizar, desacoplar y modularizar componentes

Beneficios

- Reducir el tiempo, esfuerzo y coste para ejecutar la iniciativa y evolucionar los servicios
- Aumentar la fiabilidad y escalabilidad del software.
- Facilitar el mantenimiento del software.

Precondiciones

- El equipo de desarrollo y el tech lead deben estar familiarizados con los patrones de diseño de la industria y los frameworks y librerías disponibles en el Banco y haber recibido formación sobre los mismos
- Disponibilidad de la descripción técnica de alto nivel (e.g., diseños de solución, ficha de la iniciativa, Modelo de Seguridad) de los distintos servicios y componentes de software en la herramienta de documentación global. Estos deben ser accesibles por todos los equipos siempre que no sean componentes críticos.

Adopción

2.4.1. Reutilizar, desacoplar y modularizar componentes

Los arquitectos e ingenieros de software **deberían** **reutilizar componentes** a la hora de diseñar e implementar soluciones de software.

Desde el punto de vista de diseño de nuevas iniciativas de software, los arquitectos e ingenieros **deberían** aplicar el principio de diseño de software basado en **componentes** (a.k.a. CBSE). Dicho principio se basa en la creación de **componentes reutilizables, desacoplados y modulares.**

Los componentes de software deben agruparse dentro de un servicio (descrito en la práctica 2.3) y dichos **servicios deben** cumplir con una funcionalidad de negocio acotada **evitando la duplicidad de sus componentes internos y apoyándose en otros servicios** para crear capacidades y no duplicar esfuerzo a la hora de imitar la funcionalidad que ya ofrecen otros servicios. Además, tanto los servicios como los componentes **deben** obedecer al principio de separación de intereses (**separation of concerns**).

Los Solution Architects o los ingenieros de software (especialmente el Tech Lead) **deberían** promover la implementación de patrones de arquitectura que cumplan los principios mencionados anteriormente, a continuación se mencionan **algunos de los patrones** más usados en la industria relacionados con el diseño de software:

- **Patrón de Capas (Layered Architecture)**: Este patrón divide la arquitectura de una aplicación en capas lógicas, donde cada capa tiene una responsabilidad específica. Esto

permite la reutilización de componentes a nivel de capa y facilita los cambios en la implementación de una capa sin afectar a otras capas.
- **Patrón de Microservicios:** Este enfoque arquitectónico propone construir sistemas como conjuntos de servicios pequeños e independientes que se comunican entre sí. Los microservicios pueden ser desarrollados y desplegados de forma individual, lo que facilita su reutilización y actualización independiente.
- **Patrón MVC (Modelo-Vista-Controlador):** Este patrón separa la lógica de presentación de la lógica de negocio y los datos. Al hacerlo, facilita la reutilización de las vistas y los controladores en diferentes contextos y permite cambios en la interfaz de usuario sin afectar la lógica subyacente.

Deben analizarse las necesidades de cada iniciativa y **aplicar el patrón que mejor resuelva los requerimientos funcionales y técnicos** procurando el alineamiento entre los distintos componentes de un mismo servicio (es decir, si todos los componentes de un servicio aplican el mismo patrón, entonces una nueva iniciativa dentro de ese servicio debe aplicar el patrón que ya se venía usando cuando la solución sea similar) o en cambio se acuerde un nuevo patrón para todos los componentes similares de ese servicio.

Además, los Solution Architects e ingenieros de software **deben** conocer el **stack tecnológico de el Banco** (ASO, APX, Cells, Datio, etc.) y adoptarlos según los requerimientos y necesidades de las iniciativas.

La práctica aquí recogida aplica a todas las tecnologías, siendo más factible su implementación en **sistemas nuevos** y modernos que en sistemas existentes Legacy. Aún así, los sistemas **legacy deben** evolucionar gradualmente a un modelo orientado a servicios y componentes procurando reducir el acoplamiento y la duplicidad del código de los componentes.

Herramientas

La adopción de esta práctica requiere el uso de las siguientes herramientas:

- **Herramienta de documentación global:** Herramienta pendiente de definición.

Indicadores

Los indicadores que apliquen deben ser proporcionados a los equipos para que puedan conocer su grado de madurez respecto a la práctica:

Indicador	Descripción
% de diseños de solución con design blocks	Porcentaje del total de diseños de solución almacenados en la herramienta de documentación global que contienen design blocks

Enlaces

- CBSE
- Model View Controller
- Microservicios
- Layered Architecture

2.5. Garantizar que la seguridad está implícita en el diseño de soluciones

Beneficios

- Implementar mecanismos de seguridad en una fase más temprana del proceso de desarrollo del software (shift-left). Cuanto más tarde se detecte un fallo de seguridad en el ciclo de vida del desarrollo, más costosa será su reparación y mayor su impacto al negocio.
- Ayudar a prevenir vulnerabilidades de riesgo crítico y alto, lo que permitirá reducir el número de incidentes de seguridad y su impacto.
- Estandarizar los criterios de seguridad para los proyectos y sus evolutivos, aumentando la eficiencia a través de un triaje que podrá ser evolucionado.

Precondiciones

- Formación a los equipos sobre el proceso de análisis y diseño.

Adopción

2.5.1. Garantizar que la seguridad está implícita en el diseño de soluciones

El Tech Lead o Seguridad en Proyectos, en función del modelo de participación de seguridad de la iniciativa, **deben** identificar las medidas técnicas de seguridad a aplicar en el contexto de la arquitectura de la iniciativa para proporcionar una protección adecuada frente a amenazas y cumplir requisitos regulatorios, sin perder de vista la usabilidad, operabilidad y plazos de puesta en producción.

Al comienzo de toda iniciativa, el Tech Lead **debe** llevar a cabo una evaluación de amenazas y riesgos a través del proceso de admisión de riesgos definido por el Banco, para identificar las vulnerabilidades a las que el sistema pueda hacer frente y valorar su impacto. Para ello, el Tech Lead **debe**:

- Identificar los activos (data, applications, hardware, and personnel) que necesitan protección.
- Identificar las potenciales amenazas que puedan dañar los activos (p.ej., desastres naturales, ciberataques, agresiones físicas o errores humanos).
- Determinar la probabilidad de que una amenaza se materialice.
- Determinar el potencial impacto de cada amenaza sobre cada activo.
- Determinar qué controles existen para proteger a los activos, (p.ej., medidas de seguridad físicas, controles de software, o políticas y procedimientos).
- Identificar lagunas en los controles existentes que puedan exponer a los activos a amenazas y determinar qué controles adicionales son necesarios para protegerlos de las amenazas identificadas.

En la elaboración del diseño de seguridad se **deben** analizar y tener en cuenta los siguientes aspectos dependiendo de la iniciativa (p. ej., aquellas que incluyan cambios de infraestructura, nuevas arquitecturas) para garantizar la seguridad y reflejar medidas de seguridad en el diseño:

- **Seguridad en backends (bases de datos, etc)**. Dentro de este apartado no sólo se trata del bastionado de la base de datos, sino también del esquema de autenticación, trazas de auditoría, garantizar el AAA,...
- **Seguridad en middleware (API, etc)**. Cómo se garantiza que el middleware es seguro, que su acceso está controlado a los orígenes permitidos, que utilizan mecanismos de autenticación y autorización robustos, que las claves son gestionadas de manera segura por los consumidores, que se monitorizan los aspectos de seguridad, etc.
- **Seguridad en frontales (webs, apps, etc)**. Cómo se garantiza que el acceso al frontal es seguro, la autenticación y autorización de los usuarios, integración con soluciones de IdP,
- **Seguridad en el código fuente.** Controles de seguridad en el código fuente de las aplicaciones o del código generado.
- **Seguridad en comunicaciones.** Cómo se garantiza que la información en tránsito es segura, utilizando mecanismos de seguridad apropiados y garantizando la autenticación en servidores, teniendo en cuenta el principio de zero-trust.
- **Seguridad en datos.** Garantizar la seguridad de los datos que la iniciativa gestiona.
- **Monitorización de seguridad.** Integración de los activos de la iniciativa dentro de las capacidades de "detect and respond" de el Banco.
- **Gestión de la identidad.** Gestión de los flujos de aprovisionamiento y baja de usuarios de la iniciativa. Cómo la identidad es consumida (autenticación) y el esquema de autorización.
- **Vulnerabilidades**. Garantizar que los activos que la iniciativa necesite tengan en cuenta la gestión de vulnerabilidades antes de la puesta en producción así como en BAU.

El Tech Lead **no debe** dar por concluido el diseño de seguridad de una iniciativa bajo participación Security Review hasta que no haya sido validado por Seguridad en Proyectos. De la misma manera, **no debe** dar por concluido el diseño de seguridad de una iniciativa bajo participación Specialist hasta la publicación del Modelo de Seguridad asociado.

Seguridad en Proyectos **debe** realizar periódicamente catas de los proyectos ejecutados, con el objetivo de supervisar la correcta asignación de los modelos de participación de seguridad a las iniciativas.

Herramientas

La adopción de esta práctica requiere el uso de las siguientes herramientas:

- **Herramienta de documentación global:** Herramienta pendiente de definición.
- **Formulario de Categorización** (Formulario de Admisión de Riesgos IT y Decision Framework)

Indicadores

Los indicadores que apliquen deben ser proporcionados a los equipos para que puedan conocer su grado de madurez respecto a la práctica:

Indicador	Descripción
Cata de proyectos	Revisión aleatoria de iniciativas para verificar la correcta asignación del modelo de participación de seguridad en base a los criterios definidos por Seguridad en Proyectos. Este indicador medirá el porcentaje de iniciativas en la cata a las que se ha asignado un modelo de participación de seguridad adecuadamente.

Enlaces

Site de Seguridad en Proyectos

3. Proceso

El proceso de análisis y diseño es la **etapa final de la fase de ideación** de una iniciativa. Esta fase se compone de tres etapas:

- **Detección de Oportunidades** - identificación temprana por parte de los distintos Portfolios de aquellas iniciativas que merecen ser trabajadas.
- **Review** - integración ordenada y priorizada de las iniciativas (a través de un Definition of What claro) al portfolio de proyectos de cada dominio
- **Análisis y Diseño** - diseño de la solución a construir más adecuada, incorporando desde etapas tempranas a los perfiles especialistas necesarios para ello

Una vez completada la etapa de análisis y diseño el ciclo de vida del desarrollo de software continúa con la fase de ejecución de la iniciativa.

Este capítulo describe el **proceso de análisis y diseño** alineado con las prácticas definidas en el apartado anterior. Se han descrito las responsabilidades de los **roles** involucrados en el proceso en el documento enlazado.

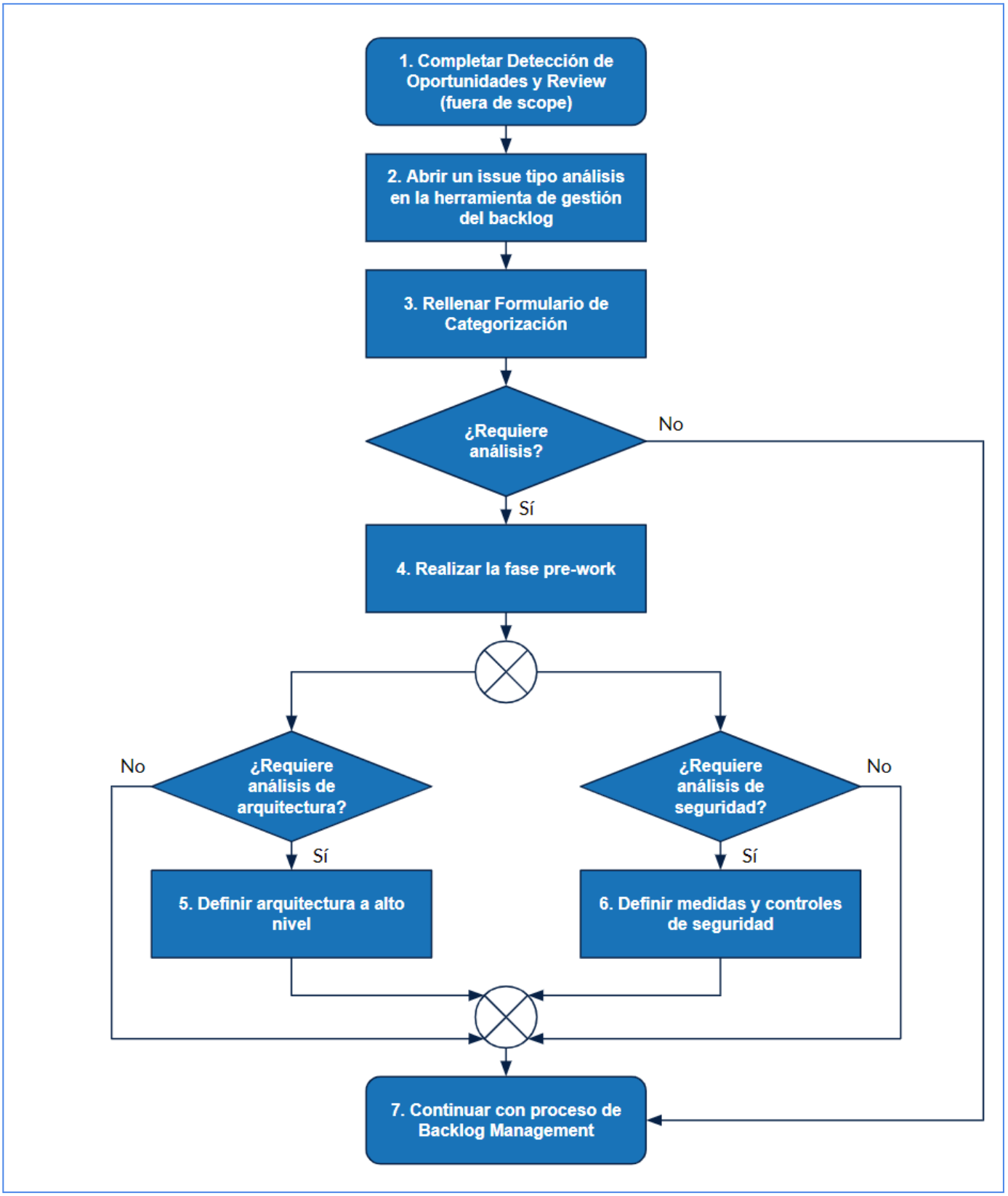


Figura 3 - Proceso de análisis y diseño

Paso	*Entregable*
1. Completar Detección de Oportunidades y Review	-
2. Abrir un issue de análisis en herramienta de backlog	-
3. Rellenar Formulario de Categorización	Formulario de Categorización (FC)
4. Realizar la fase pre-work	Ficha de la Iniciativa
5. Definir arquitectura a alto nivel	Diseño de Solución
6. Definir medidas y controles de seguridad	Modelo de Seguridad
7. Continuar con proceso de Backlog Management	-

Figura 4 - Entregables del proceso de análisis y diseño

A continuación se describen cada uno de los pasos identificados en el proceso de análisis y diseño reflejados en el diagrama anterior.

1. Completar Detección de Oportunidades y Review (fuera del scope del Playbook)

- Una vez el Project Sponsor ha identificado una iniciativa a ejecutar, se ha definido su alcance y se ha estimado los recursos necesarios **con el apoyo del Tech Lead y los perfiles especialistas relevantes**, el Project Sponsor podrá presentar la iniciativa al ciclo de priorización y aprobación de la SDA correspondiente a su geografía para ser priorizada y financiada.

- En este punto el Tech Lead también debe comunicar a los Service Owners de los servicios afectados por la iniciativa el alcance de la misma. El Tech Lead podrá apoyarse en el conocimiento del Service Owner sobre el servicio afectado a lo largo del proceso de análisis a la vez que el Service Owner se podrá apoyar en el conocimiento del Tech Lead para facilitar la planificación de su servicio.

2. Abrir un issue tipo análisis en la herramienta de gestión del backlog

- Una vez aprobada la iniciativa en la SDA, el Tech Lead debe dar de alta un nuevo **issue tipo análisis** en la herramienta de gestión del backlog para la iniciativa **para cada uno de los servicios afectados por la misma**.

3. Determinar si la iniciativa requiere un análisis y el modelo de participación de Solution Architects y Seguridad en Proyectos

- El Tech Lead debe rellenar el **Formulario de Categorización (FC)**, apoyándose en los perfiles especialistas que considere necesarios. Este formulario consta de tres partes:
    - **Criterio de análisis** - determina si se debe realizar un análisis de arquitectura o seguridad para la iniciativa. Para más información sobre el criterio para determinar si se requiere un análisis, ver la Práctica 2.1.
    - **Formulario de Admisión de Riesgos IT (FARIT)** - identifica el nivel de riesgo de la iniciativa en relación a Technology Security, Information & Data Security y Digital Fraud.
    - **Decision Framework (DF)** - determina el nivel de participación de Solution Architects y Seguridad en Proyectos en el proceso de análisis de la iniciativa cuando un análisis sea necesario. Para más información sobre los modelos de participación, ver la Práctica 2.2.

  La primera y segunda parte del formulario debe rellenarse siempre mientras que la tercera sólo debe completarse si se ha determinado que un análisis es necesario.

- Si la nueva iniciativa requiere un análisis, se debe continuar el proceso con el punto 4 para comenzar la fase de pre-work. Si la iniciativa no requiere un análisis, el Tech Lead debe continuar con el proceso de Gestión del backlog.

4. Realizar la fase pre-work

- Después de rellenar el Formulario de Categorización, el Tech Lead debe crear una **Ficha de la Iniciativa**.

- El Tech Lead debe describir la iniciativa, sus objetivos, beneficios y requerimientos funcionales en la Ficha de la Iniciativa para permitir la trazabilidad del diseño que se creará y la participación del Solution Architect y Seguridad en Proyectos. En función del nivel de participación asignado y la complejidad de la iniciativa, el nivel de detalle se ajustará y se acordará entre el Tech Lead y el Solution Architect. Para más información sobre el nivel de detalle a aportar en la Ficha de la Iniciativa, ver la Práctica 2.1.

5. Definir una arquitectura de alto nivel

- El proceso a seguir para definir la arquitectura de alto nivel de la solución tras completar la fase de pre-work depende del modelo de participación del Solution Architect determinado en el punto 3.

- Si ya existe un Diseño de Solución para el servicio y/o el aplicativo impactado por la iniciativa, el diseño de arquitectura de la iniciativa debe plasmarse sobre el mismo Diseño de Solución del servicio (si este se viera afectado) y del aplicativo (si por la complejidad del servicio este nivel de detalle fuera necesario) en la herramienta de documentación global, generando una nueva versión. Si este Diseño de Solución no existiera por ser un nuevo servicio o aplicativo, se debe crear un nuevo Diseño de Solución asociándolo al proyecto en la herramienta de documentación global. Para más detalle sobre la jerarquía de la documentación en la herramienta de documentación global, ver la Práctica 2.1.

- En el caso de iniciativas globales que estén siendo desplegadas a las distintas geografías, el Tech Lead o Solution Architect debería contactar con el arquitecto autor del modelo global para:
    - Asegurar el entendimiento del modelo global
    - Comunicar cualquier modificación que sea necesaria en el modelo global para disponibilizar la solución en la geografía (p.ej. por requisitos regulatorios locales, por componentes globales que no estén disponibles y no sea factible disponibilizar en un corto período de tiempo y que impliquen adaptaciones de componentes o interfaces globales o por integraciones con componentes locales)

5.1 Modelo Go systems

- El Tech Lead debe revisar en la herramienta de documentación global los Diseños de Solución existentes del servicio y unidad aplicativa afectadas por la iniciativa, si estos existieran.

- El Tech Lead debe definir los componentes funcionales necesarios en cada capa de la arquitectura en un diseño de solución y la interconexión entre los mismos, priorizando la simplicidad y modularidad y asegurando que los componentes propuestos están alineados con las recomendaciones vigentes de la comunidad de arquitectos.

- El Tech Lead debe identificar si la solución va a ser diseñada con frameworks o arquitecturas propios de el Banco (p. ej., Cells, ASO, APX, Datio) y reflejarlo en la arquitectura.

- En aquellos casos en los que puedan existir distintas alternativas de Diseño de Solución con impacto en aspectos como plazos, esfuerzo estimado, etc., el Tech Lead debe estar en comunicación con el Project Sponsor de cara a revisar conjuntamente los pros y contras de cada una de las alternativas y acordar qué solución es la más adecuada.

5.2 Modelo Specialist

- El Solution Architect debe comprobar que el Tech Lead ha proporcionado el nivel de detalle suficiente en la Ficha de la Iniciativa. En caso contrario, el Solution Architect debe informar al Tech Lead sobre qué información falta y el Tech Lead debe proporcionarla.

- El Solution Architect y el Tech Lead deben revisar en la herramienta de documentación global los diseños de solución existentes del servicio y unidad aplicativa afectadas por la iniciativa, si estos existieran.

- El Solution Architect y el Tech Lead deben definir los componentes funcionales necesarios en cada capa de la arquitectura en un diseño de solución y la interconexión entre los mismos, priorizando la simplicidad y modularidad y asegurando que los componentes propuestos están alineados con las recomendaciones vigentes de la comunidad de arquitectos.

- El Solution Architect y el Tech Lead deben identificar si la solución va a ser diseñada con frameworks o arquitecturas propios de el Banco (p. ej., Cells, ASO, APX, Datio) y reflejarlo en la arquitectura.

- En aquellos casos en los que puedan existir distintas alternativas de Diseño de Solución con impacto en aspectos como plazos, esfuerzo estimado, etc., el Tech Lead y el Solution Architect deben estar en comunicación con el Project Sponsor de cara a revisar conjuntamente los pros y contras de cada una de las alternativas y acordar qué solución es la más adecuada.

6. Definir medidas y controles de seguridad

- El proceso a seguir tras completar la fase de pre-work para definir las medidas y controles de seguridad de la iniciativa depende del modelo de participación de Seguridad en Proyectos determinado en el punto 3.

- Si ya existe un modelo de seguridad para el servicio y/o el aplicativo, las medidas y controles de seguridad de la iniciativa deben plasmarse sobre el mismo Modelo de Seguridad del servicio (si este se viera afectado) y del aplicativo (si por la complejidad del servicio este nivel de detalle fuera necesario) en la herramienta de documentación global, generando una nueva versión. Si este modelo de seguridad no existiera por ser un nuevo servicio o aplicativo, se debe plasmar sobre un nuevo Modelo de Seguridad asociándolo al proyecto en la herramienta de documentación global. Para más detalle sobre la jerarquía de la documentación en la herramienta de documentación global, ver la Práctica 2.1.

- En el caso de iniciativas globales que estén siendo desplegadas a las distintas geografías, el Tech Lead o Seguridad en Proyectos debería contactar al autor del Modelo de Seguridad global para:
    - Asegurar el entendimiento del modelo
    - Comunicar cualquier modificación que sea necesaria en el modelo global para disponibilizar la solución en la geografía

6.1 Modelo Security Review

- Seguridad en Proyectos debe comprobar que el Tech Lead ha proporcionado el nivel de detalle suficiente en la Ficha de la Iniciativa. En caso contrario, debería informar al Tech Lead sobre qué información falta y el Tech Lead debe proporcionarla.

- El Tech Lead debe organizar una reunión con Seguridad en Proyectos, y en ella Seguridad en Proyectos debe establecer las directrices de seguridad que deberían seguirse para crear el diseño de seguridad.

- El Tech Lead debe definir las medidas de seguridad adecuadas a los requisitos de seguridad del sistema que han sido identificados en el proceso de admisión de riesgos y a las observaciones que Seguridad en Proyectos le haya proporcionado en la fase de ideación (p. ej. confidencialidad, integridad, disponibilidad).

- El Tech Lead debe plasmar las medidas y controles de seguridad en el diseño de solución del servicio (y del aplicativo, si existiera), incluyendo los controles y procesos de prueba necesarios. Para más detalle sobre los aspectos que el Tech Lead debe considerar ver la Práctica 2.5.

- Seguridad en Proyectos debe revisar el diseño de seguridad elaborado por el Tech Lead y validarlo conforme a su cumplimiento de las directrices de seguridad definidas. Si no se cumplen las exigencias de seguridad, Seguridad en Proyectos debe comunicar al Tech Lead

qué cambios deben realizarse, y el Tech Lead debe incorporar los cambios y presentar el nuevo diseño técnico a Seguridad en Proyectos para su revisión. Seguridad en Proyectos y el Tech Lead deben repetir este ciclo hasta que se cumplan las directrices de seguridad.

- Seguridad en Proyectos debe actualizar el Modelo de Seguridad del servicio, si existiera, en base al diseño de seguridad acordado con el Tech Lead.

6.2. Modelo Specialist

- Seguridad en Proyectos debe comprobar que el Tech Lead ha proporcionado el suficiente nivel de detalle en la Ficha de la Iniciativa. En caso contrario, debe informar al Tech Lead sobre qué información falta y el Tech Lead debe proporcionarla.

- Seguridad en Proyectos debe generar una nueva versión del modelo de seguridad del servicio y/o aplicativo, si este existiera, reflejando el impacto de la iniciativa en la seguridad del servicio. Si este modelo de Seguridad no existiera por ser un nuevo servicio, Seguridad en Proyectos debe crear un nuevo modelo de seguridad para el servicio.

- El Tech Lead debe validar que el modelo de seguridad propuesto por Seguridad en Proyectos es compatible con el objetivo y el diseño técnico de la iniciativa. Si no se cumplen las exigencias, el Tech Lead debe acordar con Seguridad en Proyectos qué cambios deben realizarse, y Seguridad en Proyectos debe incorporarlos y presentar el nuevo diseño de seguridad al Tech Lead para su revisión. Seguridad en Proyectos y el Tech Lead deben repetir este ciclo hasta que el diseño de seguridad propuesto sea compatible con el objetivo y el diseño técnico de la iniciativa.

- El Tech Lead no debe dar por finalizada la fase de análisis y diseño de la iniciativa hasta disponer de un Modelo de Seguridad liberado por Seguridad en Proyectos.

7. Continuar con proceso de Backlog Management

- El equipo de desarrollo debe continuar con la fase de definición de tareas y arranque del proyecto (una vez finalizada la fase de inception/brainstorming) del proceso de gestión de backlog, descrito en el Playbook de Gestión del backlog.

- Durante la fase de ejecución de la iniciativa, el equipo de desarrollo puede identificar que se requieren modificaciones en el Diseño de la Solución o el Modelo de Seguridad generados. Si se debe a un cambio de requerimientos será necesario abrir un nuevo análisis. Si se debe a otro motivo (p.ej. evolución de arquitecturas, falta de nivel de detalle en los documentos),

el Tech Lead debe **reabrir** el issue tipo análisis en la herramienta de gestión del backlog que ya existía, volviendo al paso 2 de este capítulo, continuando con los pasos subsiguientes y actualizando todos los entregables que ya se habían generado previamente para esa iniciativa.

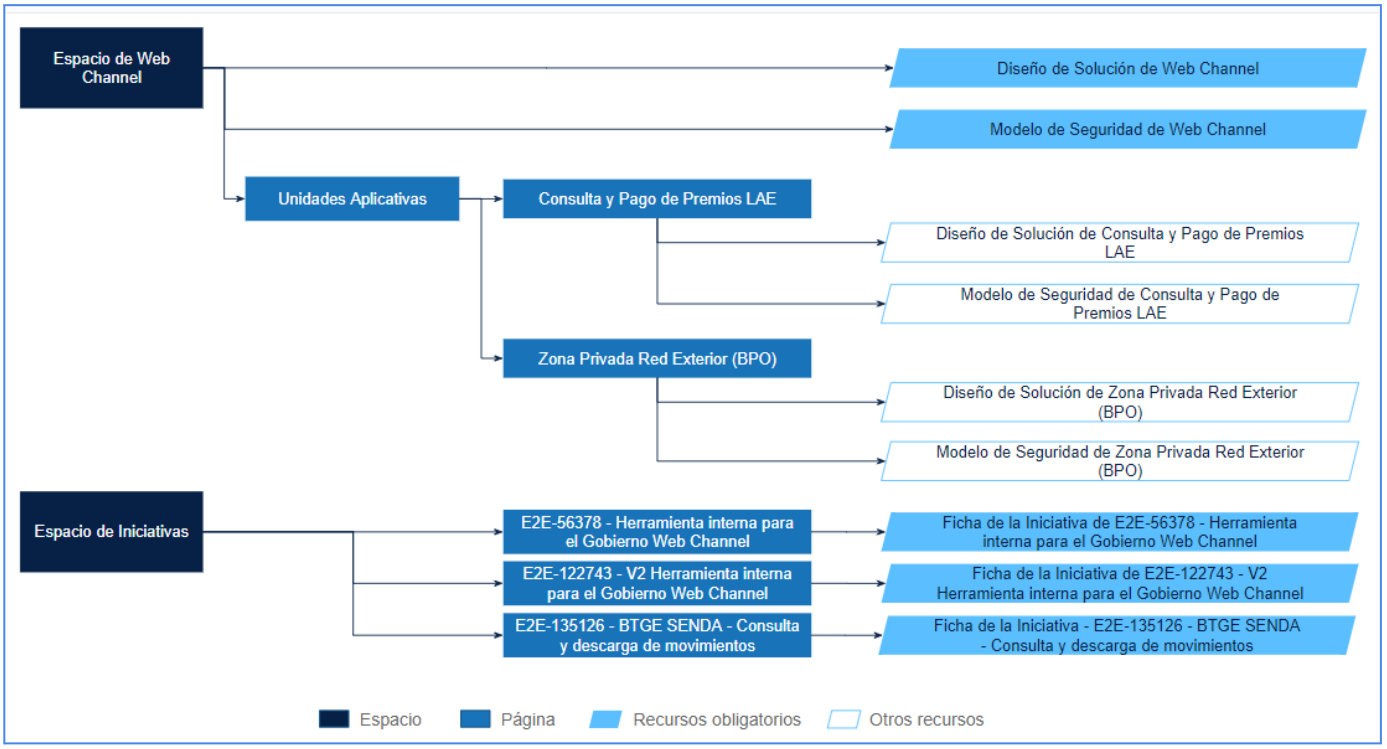
Anexo

Criterio de análisis de Seguridad (anteriormente llamado modelo Not Required)

Si la iniciativa cumple alguna de estas características, no se requiere un análisis de seguridad:

- Es PoC (con datos no reales), aislada (no hay comunicación con sistemas en el plano de red de producción) y no se va a promocionar a otros entornos (ya sean previos o productivos).
- La iniciativa dispone de un modelo de seguridad y el alcance no implica modificaciones (no hay nuevas piezas de arquitectura, no se modifican los journeys de seguridad, no hay nuevas comunicaciones a sistemas externos, ni se incluyen nuevas tipologías de datos en las cesiones existentes). Por ejemplo: proyecto de una nueva aplicación Nácar que no requiera desarrollo de nuevos servicios.
- Cambio menor en el código derivado de una incidencia o consecuencia de la realización de pruebas de calidad.
- Proyectos sin impacto funcional ni técnico en sistemas existentes, como por ejemplo:
    - Cambios de experiencia de usuario sin incluir lógica de negocio, ni nuevos consumos de servicios.
    - Aplicación de nuevos valores en la parametrización.
    - Cambios en textos de comunicaciones que no incluyan información de clientes.
- Proyectos orientados a incrementar la infraestructura debido al crecimiento vegetativo en su uso.
- Actualización de versión de productos sin necesidad de nueva infraestructura ni cambios en la arquitectura.
- Iniciativas ofimáticas.
- Evolutivos menores:
    - Modificación de una funcionalidad existente en la plataforma Host o Ether (evolucionar el backend aplicativo) sin necesidad de aprovisionamiento de datos de otras plataformas.
    - Optimización de transacciones.
    - Cambios de servicio sin necesidad de versionado o versionado de servicio que no requiera consumo de información sensible.

Ejemplo de jerarquía para el servicio Web Channel



Ejemplos de entregables

Se definen a continuación ejemplos **ilustrativos** de los documentos descritos a lo largo del playbook, al igual que la herramienta de documentación global en la que deben ser almacenados:

- Formulario de Categorización:
    - Criterio de análisis (pendiente de definición en la fase de implementación del Playbook)
    - Formulario de Admisión de Riesgos IT (FARIT)
    - Decision Framework (las preguntas del DF se revisarán en la fase de implementación del Playbook)
- Ficha de la iniciativa
- Diseño de Solución
- Modelo de Seguridad (pendiente de definición en la fase de implementación del Playbook ya que los ejemplos actuales se establecen a nivel iniciativa y no a nivel de servicio/aplicativo))
- Herramienta de documentación global (Confluence). Pendiente implantación

GLOSARIO

Análisis: Proceso de elaboración de la 1) definición funcional, 2) diseño y definición de la arquitectura a alto nivel y 3) definición del modelo de seguridad de la solución para una iniciativa de desarrollo de software.

Diseño de Solución: documento que recoge a alto nivel la definición funcional y técnica de de un servicio o aplicativo.

Equipo: Conjunto de personas que trabaja sobre un mismo backlog de tareas asociado a un producto de software.

Ficha de la iniciativa: Documento que contiene la descripción de la nueva iniciativa. Los puntos a incluir en el documento corresponden a los niveles 1, 2 y 3 del Documento de Análisis realizado hasta ahora y deberán acordarse entre el Tech Lead, Solution Architect y Seguridad en Proyectos.

Formulario de Categorización: cuestionario conformado por tres partes:

- **Criterio de análisis** – breve formulario obligatorio que determina si debe realizarse un análisis de arquitectura y de seguridad en base a las características de la iniciativa. Si el criterio de análisis indica que no es necesario un análisis de arquitectura o de seguridad, no **debe** realizarse un nuevo diseño de solución o modelo de seguridad, respectivamente.
- **Formulario de Admisión de Riesgos IT (FARIT)** – formulario de riesgo tecnológico obligatorio que determina el nivel de riesgo de la iniciativa en las dimensiones de Technology Security, Information & Data Security y Digital Fraud. El resultado del FARIT será uno de los factores utilizados por el Decision Framework para determinar el modelo de participación de Solution Architect y Seguridad en Proyectos en el análisis de la iniciativa.
- **Decision Framework** – formulario que determina el modelo de participación de Solution Architect o Seguridad en Proyectos en el análisis de la iniciativa. Este formulario solo debe cumplimentarse si el criterio de análisis determina que es necesario un análisis de arquitectura o de seguridad.

Modelo de Seguridad: documento que recoge el conjunto de controles y medidas de seguridad que deben aplicar a un servicio o aplicativo.

Project Sponsor: Perfil de negocio impulsor de una iniciativa. Este la identifica y presenta al ciclo de la SDA para su financiación.

SDA (Single Development Agenda): Cuerpo dentro del el Banco que revisa, prioriza y aprueba propuestas de iniciativas trimestralmente.

Servicio: Productos de software (p.ej., aplicación web, aplicación móvil, librerías de transacciones).