Plantilla 

January 2023

**Acerca de arc42**

arc42, La plantilla de documentación para arquitectura de sistemas y de software.

Por Dr. Gernot Starke, Dr. Peter Hruschka y otros contribuyentes.

Revisión de la plantilla: 7.0 ES (basada en asciidoc), Enero 2017

© Reconocemos que este documento utiliza material de la plantilla de arquitectura arc42, <https://www.arc42.org>. Creada por Dr. Peter Hruschka y Dr. Gernot Starke.

La versión de esta plantilla contiene textos de ayuda y explicación Se utiliza para familiarizarse con arc42 y comprender sus conceptos. Para la documentación de su propio sistema puede utilizar la version *plain*.

# Introducción y Metas

Describe los requerimientos relevantes y las directrices que los arquitectos de software y el equipo de desarrollo deben considerar. Entre estas se incluyen:

* Objetivos empresariales subyacentes, características esenciales y requerimientos funcionales para el sistema
* Metas de calidad para la arquitectura
* Las partes interesadas pertinentes y sus expectativas

## Vista de Requerimientos

Los requisitos para este proyecto se basan en crear una aplicación que permita a los turistas buscar un servicio de mapas que muestra opciones de transporte, zona de ocio, restaurantes, etc. La aplicación está pensada para que sirva de apoyo al turismo en el contexto de las ciudades inteligentes.

## Metas de Calidad

|  |  |
| --- | --- |
| Meta | Descripción |
| Usabilidad | Queremos que los usuarios sepan manejar la aplicación y para los que no son expertos de la tecnología, les sea fácil de aprender sobre cómo funciona y sea capaz de utilizarlo. |
| Funcionalidades | Presentar, filtrar y editar los datos almacenados en la BBDD de los lugares llamativos por el usuario generados por la pipeline de la tarea 1. |
| Optimización | La app tiene que funcionar ininterrumpidamente y debe soportar un gran número de consultas en el menor tiempo posible. |

## Partes interesadas (Stakeholders)

|  |  |  |
| --- | --- | --- |
| Rol/Nombre | Contacto | Expectativas |
| Equipo de desarrollo | Reuniones de trabajo para organizar el desarrollo de la app. | Diseñar y hacer funcionar la app mediante las necesidades de los usuarios. |
| Profesor | Revisará el proyecto y dará su veredicto sobre la aplicación. También podemos preguntarle dudas por correo o por tutorías. | Ser un punto de apoyo al equipo de desarrollo, para que este consiga implementar un producto funcional. |
| Usuario | Al finalizar el desarrollo de la aplicación, los usuarios valorarán mediante reseñas sobre el producto. Con eso podemos corregir errores y actualizarlos. | Tener un producto eficaz y fácil de utilizar para ellos. |

# Restricciones de la Arquitectura

|  |  |
| --- | --- |
| Restricción | Explicación |
| Permiso de ubicación | Al descargar la app, nos pedirá si podemos acceder a su ubicación donde podemos encontrar sus propias restricciones como permitir únicamente al usar la app, todo el tiempo o nada. |
| Notificaciones | Recibir notificaciones acerca de la app. |
| No usar un lenguaje técnico | Emplear un lenguaje asequible para todas las personas independientemente de su nivel de conocimiento. |
| Buena interfaz | Tiene que ser fácil y agradable para el usuario. Permite tener más posibilidades de usar con frecuencia el producto. |
| Reseñas | Los usuarios podrán ver opiniones y puntuaciones de un restaurante, pub, etc. |

# Alcance y Contexto del Sistema

|  |  |
| --- | --- |
| Socios de comunicación | Entendimiento |
| Clientes | Utilizan de forma normal con la app realizando búsquedas de gran interés para ellos. |
|  |  |

## Contexto de Negocio

**Contenido**

La especificación de **todos** los socios de comunicación (usuarios, sistemas, …) con explicaciones de las entradas y salidas específicas del dominio o interfases. Opcionalmente puede agregar formatos específicos de dominio o protocolos de comunicación

**Motivación**

Todas las partes interesadas deben entender que datos son intercambiados con el ambiente del sistema.

**Forma**

Cualquier forma de diagramas que muestren al sistema como una caja negra y especifiquen las interfases de dominio a los socios de comunicación.

De manera alternativa (o adicional) se puede utilizar una tabla. El título de la tabla es el nombre del sistema, las tres columnas contienen el nombre del socio de comunicación, las entradas y las salidas

**<Diagrama o Tabla>**

**<optionally: Explanation of external domain interfaces>**

## Contexto Técnico

**Contenido**

Las interfases técnicas (medios de transmisión y canales) enlanzando al sistema con su ambiente. De manera adicional el mapeo de las entradas/salidas específicas del dominio a los canales, es decir, una explicación acerca de que entrada/salida utiliza cual canal.

**Motivación**

Muchas partes relacionadas realizan decisiones arquitectónicas basadas en las interfases técnicas entre el sistema y su contexto. Especialmente los diseñadores de infraestructura o hardware deciden estas interfases técnicas.

**Forma**

Por ejemplo, diagramas UML de despligue describiendo los canales a sistemas vecinos, junto con una tabla de mapeo mostrando las relaciones entre los canales y las entradas/salidas.

**<Diagrama o Tabla>**

**<Opcional: Explicación de las interfases técnicas>**

**<Mapeo de Entrada/Salida a canales>**

# Estrategia de solución

**Contenido**

Un resumen corto y explicación de las estrategias y decisiones fundamentales para la solución que le dan forma a la arquitectura del sistema. Estas incluyen:

* Decisiones tecnológicas
* Decisiones acerca de la descomposición a alto nivel de un sistema, por ejemplo, el uso de algún patrón de diseño o de arquitectura.
* Decisiones en como alcanzar metas de calidad claves
* Decisiones organizacionales relevantes, como el seleccionar un proceso de desarrollo o delegar ciertas tareas a terceros.

**Motivación**

Estas decisiones son las piedras angulares de la arquitectura. Son la base de muchas otras decisiones detalladas o reglas de implementación.

**Forma**

Realice la explicación de las deciciones clave de manera breve.

Justifique las decisiones y porque se realizaron de esa manera, basado en el planteamiento del problema, las metas de calidad y restricciones clave. Refierase a los detalles en las secciones posteriores.

# Vista de Bloques

**Contenido**

La vista de bloques muestra la descomposición estática del sistema en bloques de construcción (módulos, componentes, subsistemas, clases, interfases, paquetes, bibliotecas, marcos de desarrollo, capas, particiones, funciones, macros, operaciones, estructuras de datos,…) así como sus dependencias (relaciones, asociaciones,…)

Esta vista es obligatoria para cualquier documentación de arquitectura. Es la analogía al plano de una casa.

**Motivación**

Mantener una visión general de su código fuente haciendo su estructura comprensible de manera abstracta.

Esto permite comunicar a las partes interesades en un nivel abstracto sin entrar en detalles de implementación.

**Forma**

La vista de bloques comprende una colección jerárquica de cajas negras y cajas blancas (ver figura de abajo) y sus descripciones.



**Nivel 1** comprende la descripción de Caja Blanca del sistema en general junto con las descripciones de Caja Negra de todos los bloques contenidos.

**Nivel 2** hace zoom a los bloques de construcción del Nivel 1. Entonces contiene la descripción de Caja Blanca de los bloques de construcción selecionadas del nivel 1,junto con las descripciones de caja negra de sus bloques de construcción internas.

**Nivel 3** Hace zoom a los bloques selectos del nivel 2, y así sucesivamente.

## Sistema General de Caja Blanca

Aquí se describe la descomposición del sistema en general usando la siguiente plantilla de caja blanca. Contiene:

* Un diagrama general
* La motivación para la descomposición
* Descripciones de caja negra de los bloques de construcción contenidos. Para estos se ofrecen las siguientes alternativas:
  + Usar *una* tabla para una revisión pragmática y corta de todos los bloques de construcción contenidos y sus interfaces
  + Usar una lista de descripciones de caja negra de los bloques de construcción acorde a la plantilla de caja negra (ver abajo). Dependiendo de la herramienta utilizada, esta lista podría constar de sub-capítulos (en archivos de texto), sub-páginas (en un wiki) o elementos anidados (en una herramienta de modelado).
* (opcional:) Interfases importantes, que no están explicadas en las plantillas de caja negra de un bloque de construcción, pero que son muy importantes para entender la caja blanca. En el peor de los casos se deberá especificar y desribir la sintaxis, semántica, protocolos, manejo de errores, restricciones, versiones, calidades, compatibilidades necesarias, entre otras. En el mejor de los casos bastará con ejemplos o la firma de los mismos.

***<Diagrama general>***

Motivación

*<Explicación en texto>*

Bloques de construcción contenidos

*<Desripción de los bloques de construcción contenidos (Cajas negras)>*

Interfases importantes

*<Descripción de las interfases importantes>*

Inserte las explicaciones de las cajas negras del nivel 1:

Si usa la forma tabular solo describa las cajas negras con nombre y responsabilidad acorde al siguiente esquema:

| **Nombre** | **Responsabilidad** |
| --- | --- |
| *<caja negra 1>* | *<Texto>* |
| *<caja negra 2>* | *<Texto>* |

Si utiliza una lista de descripciones de cajas negras entonces llene una plantilla de caja negra por cada bloque de construcción importante. El título es el nombre de la caja negra.

### <Caja Negra 1>

Aqui se describe la <caja negra 1> acorde a la siguiente plantilla:

* Propósito/Responsabilidad
* Interfases, cuando no son extraídas como párrafos separados. Estas interfases pueden incluir características de calidad y rendimiento.
* (Opcional) Características de Calidad / Rendimiento de la caja negra, por ejemplo, disponibilidad, comportamiento en ejecución, …
* (Opcional) Ubicación archivo/directorio
* (Opcional) Requerimientos satisfechos (si se necesita contar con la trazabilidad a los requerimientos).
* (Opcional) Incidentes/problemas/riesgos abiertos

*<Propósito/Responsabilidad>*

*<Interfase(s)>*

*<(Opcional) Características de Calidad/Performance>*

*<(Opcional) Ubicación Archivo/Directorio>*

*<(Opcional) Requerimientos Satisfechos>*

*<(Opcional) Riesgos/Problemas/Incidentes Abiertos>*

### <Caja Negra 2>

*<plantilla de caja negra>*

### <Caja Negra N>

*<Plantilla de caja negra>*

### <Interfase 1>

…

### <Interfase m>

## Nivel 2

Aquí se especifica la estructura interna de (algunos) bloques de construcción del nivel 1 como cajas blancas.

Debe decidir cuales bloques de construcción del sistema son lo suficientemente importantes para justificar una descripción detallada. Prefiera la relevancia sobre la completitud. Especifique bloques de construcción importantes, sorprendentes, riesgosos, complejos o volátiles. Deje fuera las partes normales, simples, estándares o aburridas del sistema.

### Caja Blanca *<bloque de construcción 1>*

…Describe la estructura interna de *bloque de construcción 1*.

*<plantilla de caja blanca>*

### Caja Blanca *<bloque de construcción 2>*

*<plantilla de caja blanca>*

…

### Caja Blanca *<bloque de construcción m>*

*<plantilla de caja blanca>*

## Nivel 3

Aqui se especifica la estructura interna de (algunos) de los bloques de construcción del nivel 2 como cajas blancas.

Cuando la arquitectura requiera más niveles detallados copiar esta sección para niveles adicionales.

### Caja Blanca <\_bloque de construcción x.1\_>

Especifica la estructura interna de *bloque de construcción x.1*.

*<plantilla de caja blanca>*

### Caja Blanca <\_bloque de construcción x.2\_>

*<plantilla de caja blanca>*

### Caja Blanca <\_bloque de construcción y.1\_>

*<plantilla de caja blanca>*

# Vista de Ejecución

**Contenido**

La vista de ejecución describe el comportamiento concreto y la interacción de los bloques de construcción del sistema en forma de escenarios en las siguientes áreas:

* Casos de uso o características importantes: ¿Cómo los ejecutan los bloques de construcción?
* Interacciones en interfases externas críticas: ¿Cómo cooperan los bloques de construcción con los usuarios y sistemas vecinos?
* Administración y operación: Carga, inicialización, detención.
* Escenarios de error y excepción.

Observación: El criterio principal para la elección de los escenarios posibles (flujos de trabajo, secuencias) es su **relevancia arquitectónica**. **No** es importante describir un gran número de escenarios. Se debe documentar una selección representativa.

**Motivación**

Debe entenderse como las instancias de los bloques de construcción del sistema realizan su trabajo y se comunican en tiempo de ejecución. Deben capturarse principalmente los escenarios que comuniquen a las partes relacionadas que tengan problemas para comprender los modelos estáticos en la documentación (Vista de Bloques de Construcción, Vista de Despliegue).

**Forma**

Hay muchas notaciones para describir los escenarios, por ejemplo: \* Lista numerada de pasos (en lenguaje natural). \* Diagramas de flujo o de actividades \* Diagramas de secuencia \* BPMN o EPCs (Cadenas de procesos de eventos) \* Máquinas de estado \* ….

## <Escenario de ejecución 1>

* *<Inserte un diagrama de ejecución o la descripción del escenario>*
* *<Inserte la descripción de aspectos notables de las interacciones entre los bloques de construcción mostrados en este diagrama.>*

## <Escenario de ejecución 2>

## …

## <Escenario de ejecución n>

# Vista de Despliegue

**Contenido**

La vista de despliegue describe:

1. La infraestructura técnica usada para ejecutar el sistema, con elementos de infraestructura como locaciones geográficas, ambientes, computadoras, procesadores, canales y topologías de red así como otros elementos de infraestructura.
2. El mapeo de los bloques de construcción (software) en dichos elementos de infraestructura.

Comúnmente los sistemas son ejecutados en diferentes ambientes, por ejemplo, ambiente de desarrollo, de pruebas, de producción. En dichos casos deberían documentarse todos los ambientes relevantes.

Deberá documentarse la vista de despliegue de manera especial cuando el software se ejecute como un sistema distribuido con mas de una computadora, procesador, servidor o contenedor o cuando se diseñen los procesadores y chips de hardware propios.

Desde una perspectiva de software es suficiente con capturar los elementos de la infraestructura necesarios para mostrar el despliegue de los bloques de construcción. Los arquitectos de hardware pueden ir más alla y describir la infraestructura a cualquier nivel de detalle que requieran.

**Motivación**

El software no corre sin haardware. El hardware subyacente puede influenciar el sistema o algunos conceptos entrecruzados. Por ende, es necesario conocer la infraestructura.

**Forma**

Quizá el más alto nivel de diagrama de despliegue esté contenido en la sección 3.2. como contexto técnico con la propia infraestructura como UNA caja negra. En esta sección se deberá realizar un acercamiento a ésta caja negra utilizando diagramas de despliegue adicionales:

* UML provee diagramas de despliegue para expresar la vista. Uselos, probablemente con diagramas anidados.
* Cuando las partes relacionadas de Hardware prefieran otro tipo de diagramas además de los diagramas de despliegue, permítales usar cualquier tipo que permita mostrar los nodos y canales de la infraestructura.

## Nivel de infraestructura 1

Describa (Usualmente en una combinación de diagramas, tablas y texto):

* La distribución del sistema en múltiples ubicaciones, ambientes, computadoras, procesadores, … así como las conexiones físicas entre ellos
* La motivación o justificación de importancia para la estructura de despliegue
* Características de Calidad y/o rendimiento de la infraestructura
* El mapeo de los artefactos de software a los elementos de la infraestructura.

Para múltiples ambientes o despliegues alternativos copie esta sección para todos los ambientes relevantes.

***<Diagrama General>***

Motivación

*<Explicación en forma textual>*

Características de Calidad/Rendimiento

*<Explicación en forma textual>*

Mapeo de los Bloques de Construcción a Infraestructura

*<Descripción del mapeo>*

## Nivel de Infraestructura 2

Aquí puede incluir la estructura interna de (algunos) elementos de infraestructura del nivel 1.

Copie la estructura del nivel 1 para cada elemento elegido.

### *<Elemento de Infraestructura 1>*

*<diagrama + explicación>*

### *<Elemento de Infraestructura 2>*

*<diagrama + explicación>*

…

### *<Elemento de Infraestructura n>*

*<diagrama + explicación>*

# Conceptos Transversales (Cross-cutting)

**Contenido**

Esta sección describe de manera general, las principales ideas de solución y regulación que son relevantes en multiples partes (→ cross-cutting/transversales) del sistema. Dichos conceptos están relacionados usualmente a múltiples bloques de construcción. Pueden incluir diversos temas, tales como:

* Modelos de dominio
* Patrones de arquitectura o patrones de diseño
* Reglas de uso para alguna tecnología específica.
* Decisiones técnicas principales o generales
* Reglas de implementación

**Motivación**

Conceptos que forman la base para la *integridad conceptual* (consistencia, homogeneidad) de la arquitectura. Entonces, son una contribución importante para alcanzar la calidad interna del sistema.

Algunos de estos conceptos no pueden ser asignados a bloques de construcción individuales (por ejemplo seguridad). Este es el lugar en la plantilla provisto para una especificación cohesiva de dichos conceptos.

**Forma**

La forma puede ser variada:

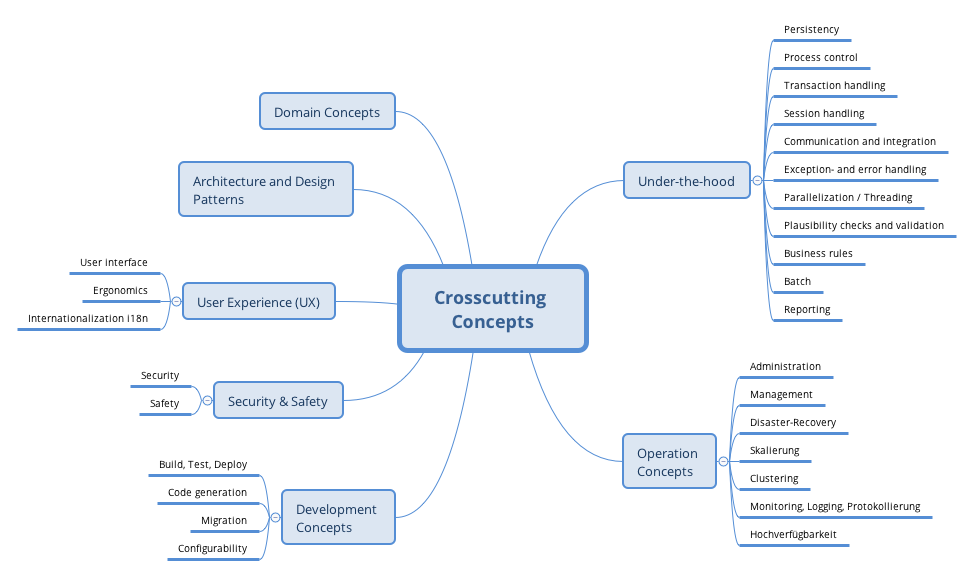
* Papeles conceptuales con cualquier tipo de estructura
* Modelo transversal (cross-cutting) de fragmentos o escenarios usando notación de las vistas arquitectónicas
* Implementaciones de muestra, especialmente para conceptos técnicos.
* Referencias a uso típico en frameworks estándar (por ejemplo, el uso de Hibernate para mapeo Objeto/Relacional) The form can be varied:

**Estructura**

La estructura potencial (pero no obligatoria) para esta sección podría ser:

* Conceptos de dominio
* Conceptos de experiencia de usuario (UX)
* Conceptos de seguridad
* Patrones de diseño y arquitectura \* A potential (but not mandatory) structure for this section could be:
* Domain concepts
* User Experience concepts (UX)
* Safety and security concepts
* Architecture and design patterns
* "Bajo el capó"
* Conceptos de desarrollo
* Conceptos de operación

Nota: Puede ser difícil asignar conceptos individuales a un tema específico de la lista



## *<Concepto 1>*

*<explicación>*

## *<Concepto 2>*

*<explicación>*

…

## *<Concepto n>*

*<explicación>*

# Decisiones de Diseño

**Contenido**

Decisiones arquitectónicas importantes, costosas, a larga escala o riesgosas incluyendo sus razonamientos. Con "Decisiones" nos referimos a la elección de una alternativa basada en cierto criterio.

Se debe usar el juicio para decidir si una decisión arquitectónica debe ser documentada en esta sección central o si sería preferible documentarla localmente (Por ejemplo, dentro de una plantilla de caja blanca de un bloque de construcción).

Evite la redundancia. Tomar de referencia la sección 4, donde ya se capturaron las decisiones más importantes para la arquitectura.

**Motivación**

Las partes relacionadas del sistema deben comprender y trazar las decisiones.

**Forma**

Varias opciones:

* Lista o tabla, ordenada por importancia y consecuencias o:
* Mayor detalle en secciones separadas por cada sección.
* Registro de Decisiones de Arquitectura (ADR por sus siglas en inglés) para cada decisión importante.

# Requerimientos de Calidad

**Contenido**

Esta sección describe todos los requerimientos de calidad como un árbol de calidad con escenarios. Los más importantes ya han sido descritos en la sección 1.2 (Metas de Calidad).

Aquí se capturan los requerimientos de calidad con menor prioridad, que no crearán altos riesgos en caso de que no sean cubiertos con totalidad.

**Motivación**

Dado que los requerimientos de calidad tendrán mucha influencia en las decisiones arquitectónicas deben tomarse en cuenta los elementos importantes para las partes relacionadas que sean concretas y medibles.

## Árbol de Calidad

**Contenido**

El árbol de calidad (Definido en ATAM - Método de análisis de compensación de arquitectura por sus silas en inglés) con escenarios de calidad/evaluación como hojas.

**Motivación**

La estructura de árbol con prioridades provee un vistazo general para un gran número de requerimientos de calidad.

**Forma**

El árbol de calidad es un vistazo a alto nivel de las metas de calidad y requerimientos:

* Un refinamiento del término de "calidad" a manera de árbol. Utilice "calidad" o "utilidad" como raíz.
* Un mapa mental con categorías de calidad como ramas principales

En cualquier caso incluya ligas a los escenarios de las siguientes secciones.

## Escenarios de calidad

**Contenido**

Concretización de requerimientos de calidad (que pueden ser vagos o implícitos) utilizando escenarios de calidad.

Estos escenarios describen lo que debería pasar cuando un estímulo llega al sistema.

Para los arquitectos, son importantes dos tipos de escenarios:

* Escenarios de uso (también llamados escenarios de aplicación o escenarios de caso de uso), que describen la reacción en tiempo de ejecución de un sistema a un determinado estímulo. Esto incluye también escenarios que describen la eficiencia o el rendimiento del sistema. Por ejemplo: El sistema reacciona a la petición de un usuario en un segundo.
* Escenarios de cambios, describen la modificación del sistema a su ambiente inmediato. Por ejemplo: Cuando se implementa funcionalidad adicional o requerimietnos para el cambio de un atributo de calidad.

**Motivación**

Los escenarios crean requerimientos de calidad concretos y permiten medirlos de manera mas sencilla o decidir si han sido cumplidos.

Cuando se requiere evaluar la arquitectura utilizando métodos como ATAM se necesitan describir las metas de calidad (de la sección 1.2) de manera más precisa hasta un nivel de escenarios que pueden ser discutidos y evaluados.

**Forma**

Texto en forma libre o tabular.

# Riesgos y deuda técnica

**Contenido**

Una lista de los riesgos técnicos o deuda técnica identificada, ordenada por prioridad.

**Motivación**

"El manejo de riesgos es administración de proyectos para gente adulta" (Tim Lister, Atlantic Systems Guild.)

Esto debiera ser el lema para la detección sistemática y la evaluación de riesgos y deuda técnica en la arquitectura, que será requerida por las partes relacionadas administrativas (por ejemplo, administradores de proyectoes, propietarios de producto) como parte de la planeación y medición de riesgos en general.

**Forma**

Lista de riesgos y/o deuda técnica, que podría incluir una medidas sugeridas para minimizar, mitigar o evitar riesgos o reducir la deuda técnica.

# Glosario

**Contenido**

Los términos técnicos y de dominio más importantes que serán utilizados por las partes relacionadas al discutir el sistema.

También se puede usar el glosario como fuente para traducciones si se trabaja en equipos multi-lenguaje.

**Motivación**

Deberían definirse claramente los términos, para que todas las partes relacionadas:

* Tengan un entendimiento idéntico de dichos términos
* No usen sinónimos y homónimos
* Crear una tabla con las columnas <Término> y <Definición>.
* Se pueden agregar más columnas en caso de que se requieran traducciones.

| Término | Definición |
| --- | --- |
| *<Término-1>* | *<definicion-1>* |
| *<Término-2>* | *<definicion-2>* |

NOTAS

Escala de tiempo

Descripción generada automáticamente con confianza baja