

19-12-2018

Documento de Arquitectura de Software (SAD)

Proyecto Comunidad Organizada



Valeria Garro Abarca
Philip Arias Ares
Manrique J. Durán Vásquez

TECNOLOGICO DE COSTA RICA

Control del documento

Proyecto: Comunidad organizada

Título: Arquitectura del Sistema –[v1.0 al 19 de dic. de 18] IEEE – 1471 – 2000

Generado por:

Estudiante de Ingeniería en Computación - [Valeria Garro Abarca]

Aprobado por:

[Acá tiene que poner la firma de la clienta]

Histórico de Revisiones

Fecha	Versión	Descripción	Autor
19/12/2018	<1.1>	Versión inicial	Valeria Garro Abarca

Tabla de Contenidos

Introducción	4
Propósito	4
Alcance	4
Usuarios Interesados	4
Recomendaciones de conformidad con esta práctica.	4
Referencias	5
Definiciones, acrónimos y abreviaciones.	5
Framework Conceptual	6
Descripción de la arquitectura en contexto	6
Stakeholders y sus roles	6
Actividades de arquitectura en el ciclo de vida	6
Usos de las descripciones de arquitectura	6
Descripciones prácticas de arquitectura	6
Documentación de la arquitectura	6
Identificación de los Stakeholders y sus responsabilidades	7
Selección de puntos de vista de la arquitectura	7
Vistas de la arquitectura	8
Vista de Escenarios	8
Vista Lógica	9
Diagrama de estados	9
Estado de Producto	¡Error! Marcador no definido.
Vista de despliegue	9
Diagrama de Componentes	9

Vista Física	10
Diagrama de despliegue	10
Vista de conceptual	11
Diagrama de conceptual	11
Vista de procesos	12
Diagramas de Actividad	12
Vista de datos	12
Diagrama de base de datos	12

1. INTRODUCCIÓN

1.1 PROPÓSITO

Este documento proporciona una arquitectónica del sistema, usando un número finito de vistas diferentes para representar los distintos aspectos que se requieren para capturar y transportar las decisiones significativas que han sido hechas sobre el sistema.

1.2 ALCANCE

El presente documento contiene el diseño elaborado para el sistema *Comunidad Organizada*, el cual es producto de un análisis minucioso de los requisitos del sistema, según estos pueden ser satisfechos con las tecnologías y características discutidas con los clientes y usuarios.

El documento está organizado alrededor de tres ideas principales:

- Las características generales del diseño.
- Los requisitos atendidos por el diseño.
- Los modelos y vistas que lo detallan.

Al contrario de muchas otras actividades técnicas, el desarrollo de sistemas de software dedica la mayoría de sus esfuerzos a la especificación y modelado.

Los modelos son utilizados tanto para el análisis de requisitos, como para el diseño de la solución, así como para la especificación, construcción y despliegue del sistema en su ambiente de explotación.

Los modelos son presentados por vistas o diagramas, generalmente utilizando notaciones gráficas como el UML.

Por otro lado, los programas de computadora son construidos por medio del uso de herramientas de traducción automáticas llamados compiladores, para los cuales es construida la forma lineal y más detallada del software del sistema: el código fuente.

Este documento ha sido generado directamente del análisis del sistema y el modelo de diseño puesto e implementado en la herramienta de diseño Lucidchart. Las secciones han sido diseñadas siguiendo el modelo de vista 4+1, también se adjunta el modelo conceptual y una vista de datos que contiene la información de la base de datos a implementar en el sistema y el diagrama de bases de datos.

1.3 USUARIOS INTERESADOS

Este documento de arquitectura va destinado a todos aquellos usuarios que estén interesados en ver y comprender el diseño y construcción de la aplicación, y sirve para que los desarrolladores de software puedan construir el bajo nivel de la aplicación usando el lenguaje que más les conviene.

1.4 RECOMENDACIONES DE CONFORMIDAD CON ESTA PRÁCTICA.

N/A.

2. REFERENCIAS

Las referencias aplicables a este documento son:

- *IEEE 830-1998 ST*
- *Architecture Tradeoff Analysis Method*
- *ISO 9126-2001 Calidad del Software y Métricas de evaluación*
- *The 4+1 View .Kruchten -1009*

3. DEFINICIONES, ACRÓNIMOS Y ABREVIACIONES.

DAS: Documento de Arquitectura de Software

Comunidad organizada: Es una aplicación de recolección y análisis de datos para la toma de decisiones.

Servidor Web: Es un programa que implementa el protocolo HTTP (HyperText Transfer Protocol). Este protocolo está diseñado para transferir lo que llamamos hipertextos, páginas web o páginas HTML (HyperText Markup Language): textos complejos con enlaces, figuras, formularios, botones y objetos incrustados como animaciones o reproductores de música.

HTTP: El protocolo de transferencia de hipertexto (HTTP, HyperText Transfer Protocol)

TCP: Protocolo de control de transmisión.

Arquitectura de Software: conjunto de elementos estáticos, propios del diseño intelectual del sistema, que definen y dan forma tanto al código fuente, como al comportamiento del software en tiempo de ejecución. Naturalmente este diseño arquitectónico ha de ajustarse a las necesidades y requisitos del proyecto.

Descripción de la Arquitectura: colección de productos de documentación.

Vistas: es una representación de un área de interés o perspectiva del sistema en alto nivel.

Tipos de Vistas: especificación de una convención de cómo construir y usar una vista. Deben satisfacer la capacidad de creación y análisis de una vista.

Stakeholder: Individuo, equipo u organización con intereses relativos al sistema.

Escenario: especifica el comportamiento y limita el interés de un área específica del sistema para uno o varios stakeholders.

4. FRAMEWORK CONCEPTUAL

4.1 DESCRIPCIÓN DE LA ARQUITECTURA EN CONTEXTO

Este documento presenta la arquitectura como una serie de vistas basadas en la arquitectura de software del modelo 4+1 de Kruchten. Estas vistas son: la vista de escenarios o +1, la vista lógica, la vista de despliegue, la vista física, la vista de procesos. No hay ninguna vista separada de una misma implementación, descrita en este documento. Estas vistas están hechas sobre Lenguaje de modelo unificado (UML) en su versión 2.0 desarrolladas usando Lucidchart.

Los estilos arquitectónicos serán referenciados en este documento de arquitectura, según las recomendaciones de la Arquitectura de software del modelo 4+1 de Kruchten.

4.2 STAKEHOLDERS Y SUS ROLES

Este documento representa la identificación de Stakeholders y sus roles a partir de la interpretación de los casos de uso o de la documentación presentada en la especificación de requerimientos.

4.3 ACTIVIDADES DE ARQUITECTURA EN EL CICLO DE VIDA

N/A.

4.4 USOS DE LAS DESCRIPCIONES DE ARQUITECTURA

Las descripciones de arquitectura de este documento se usarán para referenciar el diseño del sistema de software.

5. DESCRIPCIONES PRÁCTICAS DE ARQUITECTURA

N/A.

5.1 DOCUMENTACIÓN DE LA ARQUITECTURA

N/A.

5.2 IDENTIFICACIÓN DE LOS STAKEHOLDERS Y SUS RESPONSABILIDADES

Stakeholder	Descripción	Escenario	Vistas
Invitado	Es el usuario no registrado aun en el sistema.	Todos los escenarios descritos en el módulo invitado.	Casos de uso que van desde CU # a #
Usuario	Es la persona que interactúa con el sistema, realiza los reportes.	Todos los escenarios descritos en el módulo Usuario.	Casos de uso que van desde CU # a #
Administrador	Es la persona que interactúa con el sistema, ingresa elementos a los catálogos de incidencias, gestiona comunidades, usuarios y publicaciones..	Todos los escenarios descritos en el módulo Administrador	Casos de uso que van desde CU # a #

5.3 SELECCIÓN DE PUNTOS DE VISTA DE LA ARQUITECTURA

Vistas	UML
Escenarios (+1)	Casos de Uso
Lógica	Clases, Estados
Despliegue	Componentes, Paquetes
Física	Despliegue

Procesos	Actividad, Secuencia
Datos	Diagrama de base de datos

5.4 VISTAS DE LA ARQUITECTURA

VISTA DE ESCENARIOS

DIAGRAMAS CASOS DE USO

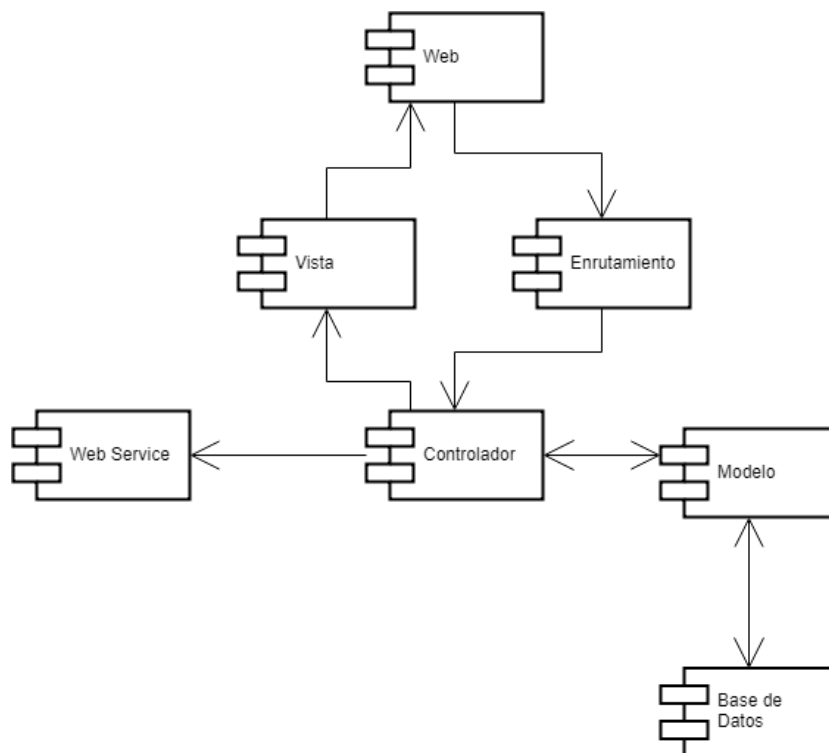


VISTA LÓGICA

DIAGRAMA DE ESTADOS

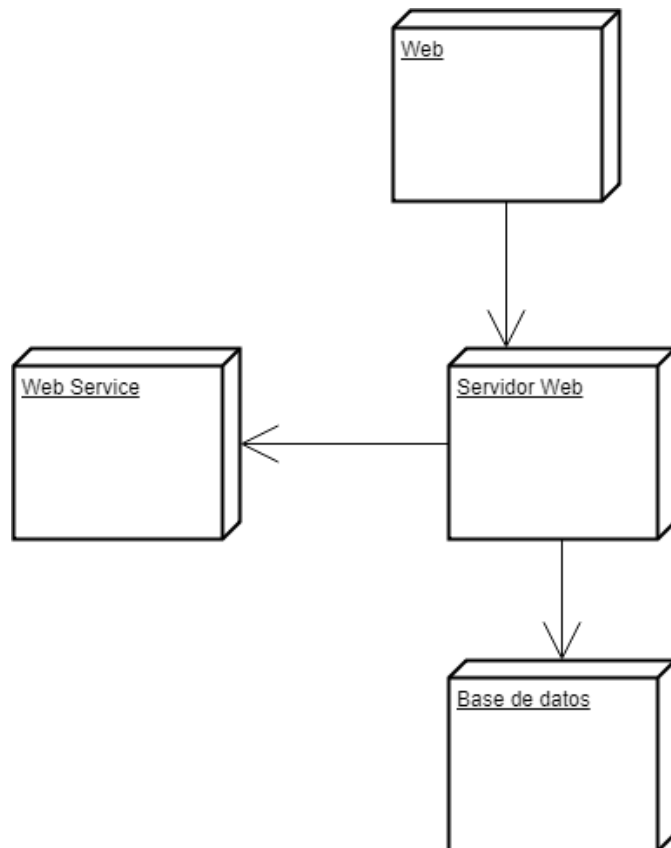
VISTA DE DESPLIEGUE

DIAGRAMA DE COMPONENTES



VISTA FÍSICA

DIAGRAMA DE DESPLIEGUE



VISTA DE CONCEPTUAL

DIAGRAMA DE CONCEPTUAL

