**Documento de Arquitectura de Software**

**IEEE-1471**

**Control del documento**

**Proyecto**

**Título**

**Generado por**

**Aprobado por**

**1. Introducción**

**1.1 Propósito**

Este documento proporciona una descripción comprensiva 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 proyecto Sistema Restaurant, 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.

1.3 Usuarios Interesados

Este documento de Arquitectura de Software (DAS), puede ser usado por todos aquellos usuarios que deseen comprender el diseño y construcción de la aplicación de Restaurant, y sirve como base para que los desarrolladores de software puedan construir el bajo nivel de la aplicación usando el lenguaje que más les acomode.

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.**

**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, la vista lógica., la vista de desarrollos, 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 IBM Rational Rose Enterprise 7.0.

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 del Negocio.

**4.3 Actividades de arquitectura en el ciclo de vida**

N/A.

**4.4 Usos de las descripciones de arquitectura**

**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**

**[EJEMPLO]**

**5.3 Selección de puntos de vista de la arquitectura.**

**[EJEMPLO]**

Vistas UML Escenarios Casos de uso Lógica Clases Desarrollo Componentes Física Despliegue Procesos Secuencia

**5.4 Vistas de la arquitectura**

**Vistas.- escenarios Diagrama.- Caso de uso del negocio – Caso de uso de diseño**

**Vista.- Lógica Diagramas.- Clases**

**Vista.- Desarrollo Diagrama de componentes general, estilo arquitectónico “N-Tiers / Orientación a objetos”**

**[EJEMPLO]**

**vista.- Física Diagrama.- Despliegue**

**[EJEMPLO]**

**Vista.- Procesos Diagrama.- Secuencia (Buscar Reserva)**

**Diagrama de secuencia (Nueva reserva)**

**Diagrama de secuencia (Registrar pago)**

**5.5 Consistencia en la cantidad de vistas de la arquitectura.**

**DESCRIPCION DE MODULOS**

**Nombre del módulo descripción Componentes inclusos**

**[EJEMPLO]**

Modulo que agrupa las

Gestion

clases e interfaces encargadas de orquestar las clases del dominio, agrupa funcionalidades que se acercan más al negocio.

Negocio

AccesoDatos

Reserva

Orden

GestionReserva

Modulo que agrupa todas las clases del negocio (Dominio), cada clase contiene su propia interfaz para exponer la funcionalidad a las otras capas, ejemplo: ReservaDom / IReservaDom. Contiene el CRUD (Create, Read, Update, Delete) del negocio.

ProveedorDom

Contiene las clases que hacen la persistencia a la base de datos, cada clase posee su propia interfaz para exponer la funcionalidad de cada tabla expresada en el modelo de datos.

MenuDao

**DESCRIPCION DE COMPONENTES**

**Nombre del componente descripción Componentes relacionados**

**[EJEMPLO]**

Contiene la lógica para: Nuevas reservas, buscar reservas, eliminar reservas, actualizar reservas.

Orden Con sus correspondientes interfaces. Contiene la lógica para: nuevas órdenes y el CRUD

**DERSCRIPCION DE CONECTORES**

**5.6 Arquitectura lógica.**

**Performances**

**[EJEMPLO]**

La arquitectura de software escogida apoya a los requerimientos no funcionales y requerimientos de arquitectura de sistemas descritos en los anexos de este documento.

1. El sistema apoyará hasta 2000 usuarios simultáneos contra la base de datos central en cualquier tiempo dado, y hasta 500 usuarios simultáneos contra los servidores locales en un momento dado.

2. El sistema proporcionará el acceso a la base de datos de catálogo de curso de herencia sin más que una 10 segunda latencia.

3. El sistema debe ser capaz de completar el 80 % de todas las transacciones dentro de 2 minutos.

4. La parte de cliente requerirá el espacio de disco de menos de 20 MB y la RAM de 32 MB.

**Calidad**

**5.7 Ejemplo de uso.**

N/A.

**5.8 Detalles de la implementación**

**5.8.1 Lenguajes y plataformas**