Documento de Arquitectura de Software

IEEE-1471-2000

**Control del documento**

Proyecto

CRB - Control Room Booking

Título

Arquitectura del Sistema – [CRB v1.0 al 28/09/2019]

Generado por

ISC Alicia Linares Ramírez

ISC Rodrigo Govea Mariano

ISC José Salvador Rodríguez Coyt

LI Alfonso Ochoa Legorreta

Aprobado por

1. **Introducción**
   1. **Antecedentes**

El Instituto Tecnológico de Estudios Superiores de Zamora, cuenta con aulas y laboratorios de cómputo adecuados para la impartición de clases del área de las Tecnologías de la Información, dichos laboratorios son utilizados por los alumnos y profesores diariamente, por lo que se requiere un registro del uso de los equipos de cómputo del centro, además no se cuenta con un control de los accesos, o bien, un sistema para ver cuáles computadoras están ocupadas o desocupadas.

* 1. **Propósito**

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

* 1. **Alcance**

El presente documento contiene el diseño elaborado para el proyecto CRB - Control Room Booking. El cuál es el producto del análisis minucioso de las necesidades que se presentan en el Instituto Tecnológico de Estudios Superiores de Zamora, más precisamente en el Centro de Cómputo, las cuales pueden ser satisfechas con las tecnologías.

El documento está organizado en las siguientes ideas:

* Las características generales del diseño.
* Los requisitos atendidos por el diseño.
* Los diseños y diagramas.
* Los modelos y vistas.

A diferencia de la mayoría de las actividades técnicas, el desarrollo de Software dedica la mayoría de sus esfuerzos a la especificación y modelado.

Los modelos son utilizados para analizar requisitos, así como para el diseño de la posible solución y 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 y apoyándonos de herramientas de desarrollo web como <https://app.cacoo.com> .

Para el manejo de la base de datos donde se va a guardar la información y se va a llevar el control y administración de la misma se utilizó SQL Server 2014 Management Studio, ya que es una herramienta confiable y soporta grandes cantidades de peticiones simultaneas, lo cual es la mayor deficiencia en otros manejadores de Bases de Datos.

El diseño de la interfaz gráfica será representado por la herramienta ASP .net el cual es un entorno para aplicaciones web desarrollado y comercializado por Microsoft. El cual nos permitirá construir el sitio web dinámico y/o la aplicación web.

Para el desarrollo de la programación o back end, se utilizó el framework Visual Studio junto con el lenguaje de programación C#, para la programación lógica y la funcionalidad de las distintas interfaces gráficas, así como las respectivas validaciones de todos y cada uno de los botones o links que se encuentran en las distintas pantallas mostradas.

* 1. **Usuarios Interesados**

Este documento de Arquitectura de Software, puede ser usado por el Instituto Tecnológico de Estudios Superiores de Zamora (ITESZ) para comprender el diseño y construcción de la aplicación de Control Room Booking (CRB), y que 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. **Recomendaciones de conformidad con esta práctica**

N/A

1. **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
* Arquitectura Cliente – Servidor.

1. **Definiciones, acrónimos y abreviaciones**

CRB: Control Room Booking: Control de Acceso al Centro de Cómputo.

UML: Lenguaje Unificado de Modelado.

ASP: Active Server Pages

BACK END: Es la parte en diseño de software que procesa la entrada desde el front-end.

FRONT END: En diseño de software es la parte del software que interactúa con los usuarios.

FRAMEWORK: Es una estructura conceptual y tecnológica de asistencia definida, normalmente, con artefactos o módulos concretos de software.

ARQUITECTURA DE SOFTWARE: Es un 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 ah de ajustarse a las necesidades y requisitos del proyecto.

1. **Framework Conceptual**
   1. **Descripción de la arquitectura en contexto**

Este documento presenta la arquitectura cliente servidor ya que es la más apta para la resolución de las necesidades a resolver con esta aplicación.

La arquitectura cliente servidor es un modelo de diseño de software en el que las tareas se reparten entre los proveedores de recursos o servicios, llamados servidores, y los demandantes, llamados clientes. Un cliente realiza peticiones a otro programa, el servidor quien le da respuesta.

En esta arquitectura la capacidad de proceso está repartida entre los clientes y los servidores, aunque son más importantes las ventajas de tipo organizativo debidas a la centralización de la gestión de la información y la separación de responsabilidades, lo que facilita y clarifica el diseño del sistema.

* 1. **StakeHolders y sus roles**

Este documento representa la identificación de StakeHolders y sus roles a conforme lo requerido y atendiendo las solicitudes hechas en el levantamiento de requerimientos.

* 1. **Actividades de arquitectura en el ciclo de vida**

**N/A**

* 1. **Usos de las descripciones de arquitectura**

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

1. **Descripciones prácticas de arquitectura**
   1. **Documentación de la arquitectura**
   2. **Identificación de los StakeHolders y sus responsabilidades**

|  |  |  |  |
| --- | --- | --- | --- |
| **Stakeholder** | **Descripción** | **Escenario** | **Vistas** |
| Administrador | Es el usuario encargado del centro de cómputo y está a cargo de la gestión del sistema. |  |  |
| Laboratorista | Usuario encargado de la verificación de salas y equipos de cómputo. Y reporta al administrador. |  |  |
| Auxiliar | Usuario encargado de la verificación de salas y equipos de cómputo. Reporta al Laboratorista. |  |  |
| Coordinador  de  Carrera | Usuario que puede apartar y/o liberar, una, varias o todas las salas de cómputo. |  |  |
| Profesor | Usuario que puede apartar y/o liberar una sala de cómputo o un equipo de cómputo. |  |  |
| Alumno | Usuario que puede apartar y/o liberar un equipo de cómputo. |  |  |
| Control de Pagos | Usuario encargado de recibir el efectivo y generar dinero electrónico. |  |  |

* 1. **Selección de puntos de vista de la arquitectura**
  2. **Vistas de la Arquitectura**
     1. **Diagrama.- Casos de uso del negocio**
     2. **Diagrama.- Clases**
     3. **Diagrama.- De Componentes General**
     4. **Diagrama.- Despliegue**
     5. **Diagrama.- Secuencia**
  3. **Consistencia en la cantidad de vistas de la arquitectura**

**Descripción de Módulos**

**Descripción de Componentes**

* 1. **Arquitectura Lógica**

**Performances**

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

1. El sistema apoyará hasta 3500 usuarios simultáneos contra la base de datos central en cualquier tiempo dado, y hasta 1000 usuarios simultáneos contra los servidores locales en un momento dado.
2. El sistema proporcionará el acceso a la base de datos del catálogo de equipos disponibles con 10 segundos de latencia como máximo.
3. El sistema debe ser capaz de completar el 75% de todas las transacciones dentro de 3 minutos.
4. La parte del cliente no requerirá espacio en disco ya que la aplicación será web y estará alojada en los servidores principales.
5. La alta disponibilidad de la red genera una confiabilidad de funcionamiento del sistema en un 99.9%.

**Calidad**

La arquitectura de software apoya las exigencias de calidad, como se encuentra estipulado en las especificaciones anexadas en este documento.

1. La interfaz de usuario será WEB.
2. La interfaz de usuario del sistema CRB será diseñado para la facilidad de uso y será apropiado para asegurar las normas de usabilidad universal establecidas por ISO 9126, ISO 14598 y la norma ISO/IEC 25000, así como el modelo CMMI.
3. Cada despliegue de opciones de pantalla, tendrá la ayuda en línea para el usuario. La ayuda en línea incluirá paso a paso instrucciones en la utilización del Sistema. La ayuda en Línea incluirá definiciones para términos y acrónimos.
   1. **Ejemplo de Uso**

**N/A**

* 1. **Detalles de Implementación**
     1. **Lenguajes y Plataformas**