

**DDIS\_v0.1\_2016**

**Documento de Diseño: BIO ASSITENS**

**DEV SOFT**

**Desarrollo e innovación de nuevas**

**Tecnologías.**

A**ctualizado a Junio de 2016**

bio assitens

Documento de Diseño

**HISTORIAL DE LAS REVISIONES**

| **Ítem** | **Versión** | **Fecha** | **Autor** | **Descripción** | **Estado** | **Responsable de Revisión y/o Aprobación** |
| --- | --- | --- | --- | --- | --- | --- |
| 1 | 0.1 | 10/06/2016 | CO | Versión preliminar revisada por QA | Revisado | BST |
|  |  |  |  |  |  |  |

**Autor(es):**

CO: Carluis Oyola

**Revisor(es) Institución:**

BST: Benji Santillán Torres

TABLA DE CONTENIDO

[1. INTRODUCCIÓN 4](#_Toc430547984)

[2. ESTRUCTURA FÍSICA DE LA INFORMACIÓN 4](#_Toc430547988)

[3. DISEÑO DE LA ARQUITECTURA DEL SISTEMA 4](#_Toc430547989)

[4. ESPECIFICACIÓN DE PROCESOS dEL SISTEMA 5](#_Toc430547989)

[5. ESPECIFICACIÓN DE INTERFACES DEL SISTEMA 6](#_Toc430547989)

[6. ESPECIFICACIÓN DE COMPONENTES SOFTWARE 9](#_Toc430547989)

[7. OBSERVACIONES 9](#_Toc430547989)

1. **INTRODUCCIÓN**

El presente documento presenta el diseño técnico de la aplicación BIO ASSITENT y cuenta con las especificaciones técnicas de las interfaces, la arquitectura, componentes de software y procesos que realice el sistema, además de la estructura física y de datos.

El documento de diseño tiene como objetivo principal ser una guía detallada de cómo se van a implementar los requisitos del sistema a desarrollar y de cómo el usuario final lo va a percibir sin entrar a detalles técnicos de implementación. En otras palabras, describe la funcionalidad esperada por el usuario final.

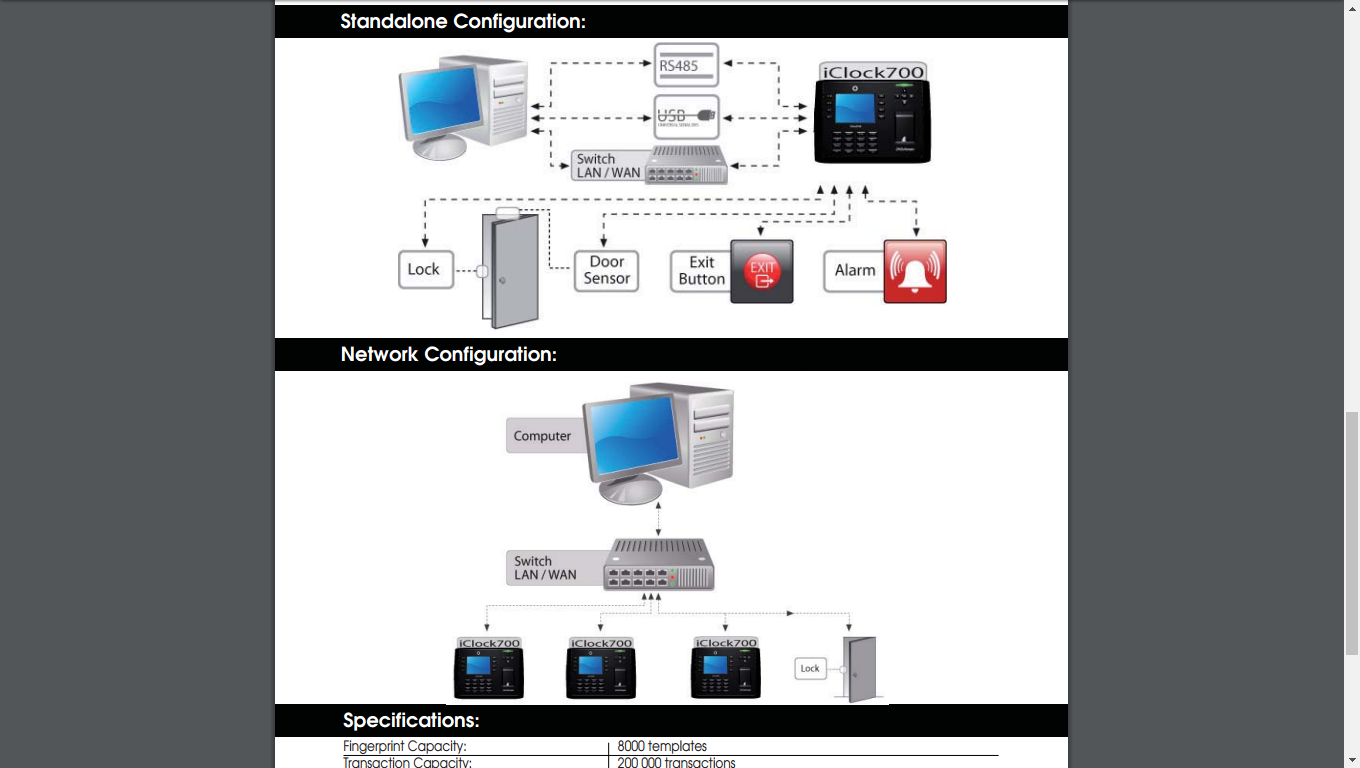
1. **OBJETIVOS Y RESTRICCIONES DE DISEÑO**

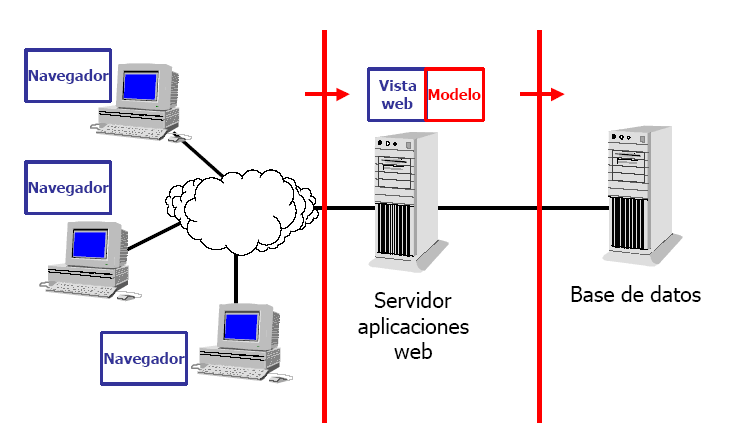
Una gran mayoría de estos objetivos y restricciones lo determinan:

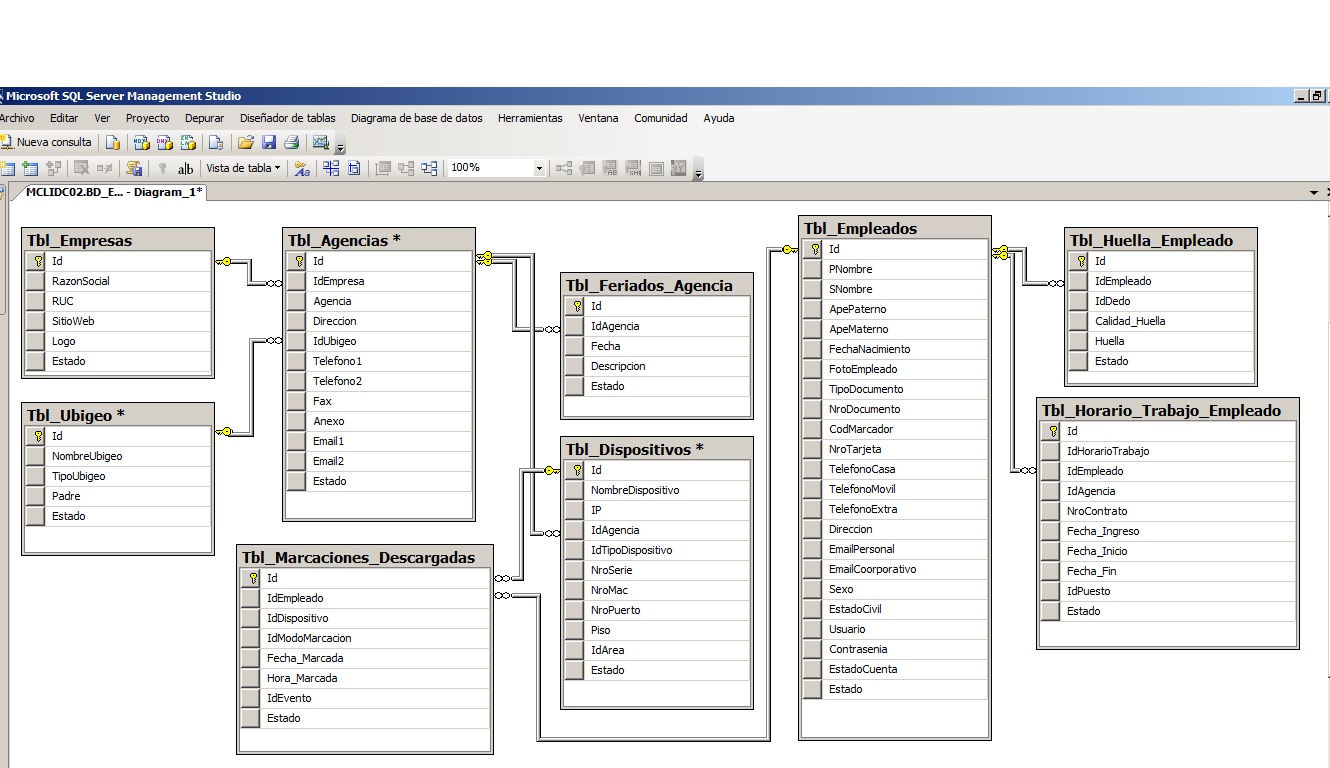
* La infraestructura del Servidor donde será instalada y/o publicada la aplicación y Base de Datos.
* El software que será utilizado para la construcción del sistema y el ambiente
* La utilización de librerías la comunicación con los relojes/marcadores.
* Los requerimientos que se le impondrán al sistema.
* Los requisitos de desempeño, seguridad, confiabilidad y calidad del producto.
* El uso de estándares y normativas que deben ser tomadas en cuenta para el desarrollo de la aplicación.

1. **ESTRUCTURA FÍSICA DE LA INFORMACIÓN**

La aplicación contará con una base de datos ubicada en uno de nuestros servidores, ya que los relojes/marcadores están conectados a través de la red configurados con un IP y un Puerto único por cada dispositivo, los datos se almacenarán directamente en la base de datos, usando dichos datos para el control y verificación del control de asistencia de los empleados en la empresa.







1. **DISEÑO DE LA ARQUITECTURA DEL SISTEMA**

La arquitectura basada en capas se enfoca en la distribución de roles y responsabilidades de forma jerárquica proveyendo una forma muy efectiva de separación de responsabilidades. El rol indica el modo y tipo de interacción con otras capas, y la responsabilidad indica la funcionalidad que está siendo desarrollada.

Por ejemplo, una aplicación web típica está compuesta por una capa de presentación (funcionalidad relacionada con la interfaz de usuario), una capa de negocios (procesamiento de reglas de negocios) y una capa de datos (funcionalidad relacionada con el acceso a datos).

El estilo de arquitectura basado en capas se identifica por las siguientes características:

* + Describe la descomposición de servicios de forma que la mayoría de la interacción ocurre solamente entre capas vecinas.
  + Las capas de una aplicación pueden residir en la misma maquina física (misma capa) o puede estar distribuido sobre diferentes computadores (n-capas).
  + Los componentes de cada capa se comunican con otros componentes en otras capas a través de interfaces muy bien definidas.
  + Este modelo ha sido descrito como una “pirámide invertida de re-uso” donde cada capa agrega responsabilidad y abstracción a la capa directamente sobre ella.
  1. **PRINCIPIOS FUNDAMENTALES**

Los principios comunes que se aplican cuando se diseña para usar este estilo de arquitectura incluyen:

* + Abstracción. La arquitectura basada en capas abstrae la vista del modelo como un todo mientras que provee suficiente detalle para entender las relaciones entre capas.
  + Encapsulamiento. El diseño no hace asunciones acerca de tipos de datos, métodos, propiedades o implementación.
  + Funcionalidad claramente definida. El diseño claramente define la separación entre la funcionalidad de cada capa. Capas superiores como la capa de presentación envía comandos a las capas inferiores como la capa de negocios y la capa de datos y los datos fluyen hacia y desde las capas en cualquier sentido.
  + Alta cohesión. Cada capa contiene funcionalidad directamente relacionas con la tarea de dicha capa.
  + Reutilizable. Las capas inferiores no tienen ninguna dependencia con las capas superiores, permitiéndoles ser reutilizables en otros escenarios.
  + Desacople. La comunicación entre las capas está basada en la abstracción lo que provee un desacople entre las capas.
  1. **BENEFICIOS**

Los principales beneficios del estilo de arquitectura basado en capas son:

• Abstracción. Las capas permiten cambios que se realicen en un nivel abstracto. Usted puede incrementar o disminuir el nivel de abstracción usado en cada capa de la “pila” jerárquica.

• Aislamiento. El estilo de arquitectura de capas permite asilar los cambios en tecnologías a ciertas capas para reducir el impacto en el sistema total.

• Rendimiento. Distribuir las capas entre múltiples sistemas (físicos) puede incrementar la escalabilidad, la tolerancia a fallos y el rendimiento.

• Mejoras en Pruebas. La capacidad de realizar pruebas se beneficia de tener una interfaces bien definidas para cada capa así como de la habilidad para cambiar a diferentes implementaciones de las interfaces de cada capa.

• Independencia. El estilo de arquitectura basado en capas el requerimiento de considerar el hardware y los problemas de instalación así como las dependencias de interfaces externas.

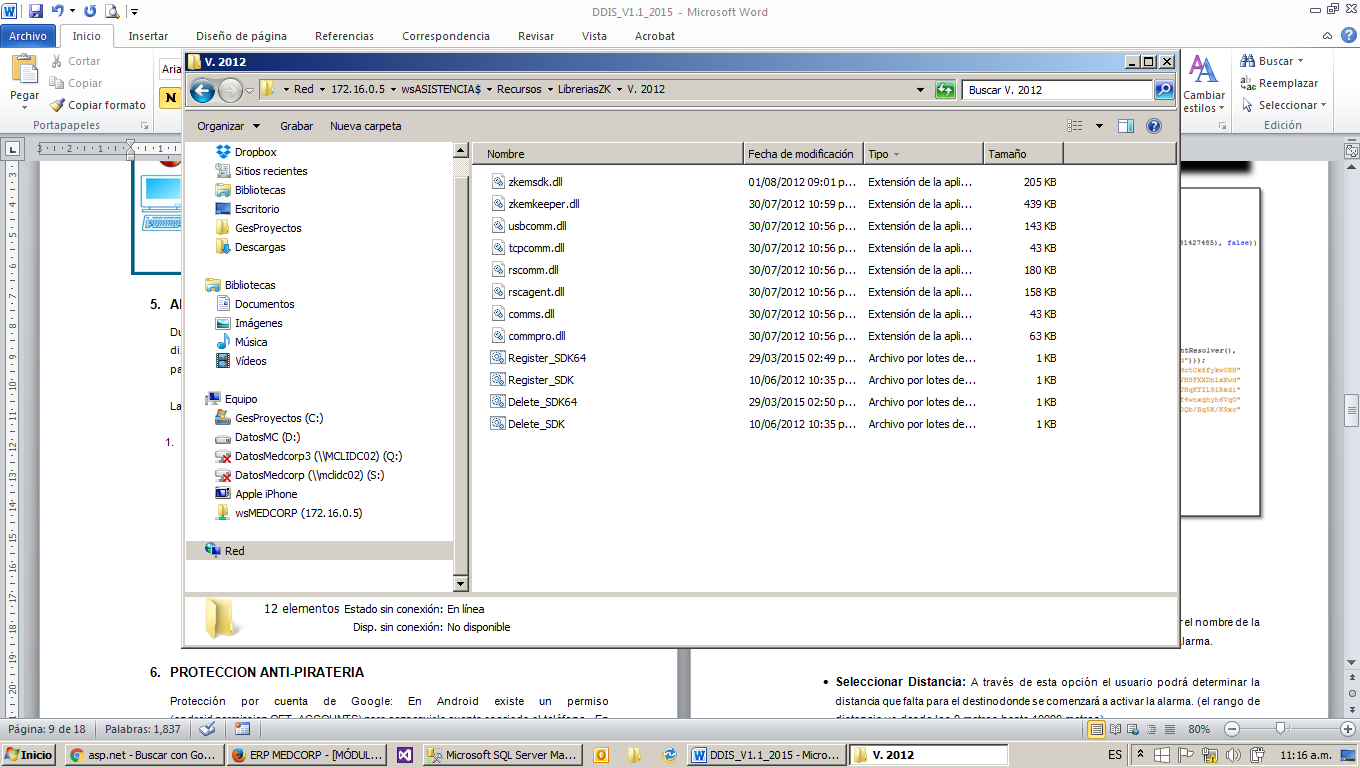


1. **ARQUITECTURA DETALLADA DE LOS MÓDULOS:**

Durante la elaboración de la aplicación se utilizó 2 librerías importantes para mejorar el diseño de la misma. Buscando de esta manera, elaborar un software amigable y llamativo para el cliente.

Las librerías usadas siguientes:

1. zkemkeeper.dll:
   1. Fuente:
   2. Versión: 2012
   3. Información: Interacción con el marcador que nos permitirá conectarnos y acceder a todas sus propiedades y funciones como son las de descargar marcaciones, huellas, subir empleados, huellas, sincronizarlos en fecha y hora, enviarle SMS.



1. **ESPECIFICACIÓN DE COMPONENTES SOFTWARE**

La aplicación solo contará con un componente de software que contendrá a los cuatro módulos previamente definidos y **detallados en el documento de análisis**, estos serán empaquetados en un folder para su posterior publicación en el servidor que cuente con las características mencionadas en el documento de Guía de Instalación y Configuración

1. **PLATAFORMA PARA EL DESARROLLO DE LA APLICACIÓN**

* **Entorno de Desarrollo Visual Studio 2012**



* **SQL SERVER 2008**



* **Componentes TELERIK UI para ASP.NET**



* **INTERNET INFORMATION SERVICES (IIS)**





* **Windows Server 2008 Enterprise Edition**



1. **OBSERVACIONES**

La aplicación no está vinculada a un servidor que permita interactuar en todo momento con los marcadores, para poder así agilizar procesos del control de asistencia de los empleados. Además de esto, la aplicación a laborar requiere que el navegador sea de la versión igual o superior a la 46.x, por lo cual quedan descartados en el funcionamiento de los navegadores cual la versión sea menor a la indicada.

|  |  |  |
| --- | --- | --- |
| *\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_* |  | *\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_* |
| *Manuel Sáenz*  *Cliente que aprueba el diseño* |  | *Carl Luis Oyola Laynes*  *Analista Funcional que elaboró el diseño* |
| *Fecha: 10/06/2016* |  | *Fecha: 10/06/2016* |