

**Dirección de Proyecto TI**

**Retroalimentación Inmediata**

**Documento de Arquitectura Software IEEE-1471**

**Ingeniería en Computación e Informática**

**Facultad de Ingeniería**

**Universidad Andrés Bello**

**Profesor Cesar Llanos**

**Documento de Arquitectura de Software IEEE-1471**

**Control del documento**

**“”**

**Proyecto:**

* *“Retroalimentación Inmediata”*

**Sprint:**

* *sprint n+1 dd/mm/aa*

**Generado por:**

* *Nombre\_Scrum*

**Aprobado por:**

* *Nombre\_SP.Documentador*

**1. Introducción**

**1.1 Propósito:** *“Para que fue hecho el documento”*

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:** EJEMPLO.

En este documento se describen las 5 vistas que representan la arquitectura del software del proyecto “NOMBRE” desde diferentes perspectivas (para el entendimiento de distintos interesados del proyecto), construidas a partir de los requerimientos y el análisis del proyecto. Además, se describen algunas características generales del diseño. A través de una vista lógica se pueden apreciar las clases involucradas en el proyecto, la vista de procesos para ver el comportamiento de las funciones del sistema, la vista de desarrollo para visualizar la interacción entre distintos componentes del sistema, la vista de despliegue para poder visualizar la distribución física del sistema y su organización, y finalmente una vista global de escenario a través de los casos de uso y ver las funcionalidades del sistema a nivel global y detallada.

**1.3 Stakeholders del documento:** *“Interesados del documento”. EJEMPLO.*

|  |  |  |
| --- | --- | --- |
| **Stakeholder** | **Area de especializacion** | **Descripción** |
| Diseñador de sistema | Diseño | Enfocado en el cómo se hará el software, desde un análisis desarrollado a un ente apreciable y visual. (A partir de requerimientos, caso de uso; se obtienen Diagramas de clases, secuencia, componentes, diálogo, etc). |
| Arquitecto de sistema | Arquitectura de software | Enfocado en la estructura, organización, comunicación y composición del sistema. De esta manera podemos establecer una base del sistema con sus componentes, su relación entre ellos, cómo se distribuyen y la integración de todos ellos |

**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. Ejemplo:**

|  |  |
| --- | --- |
| **Abreviaturas** | **Definiciones** |
| **BD** | Base de datos. |
| **IEEE** | Institute of Electrical and Electronics Engineers. |

**4. Framework Conceptual. Ejemplo:**

1. **Descripción de la arquitectura en concepto:**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.
2. **Stakeholders y sus roles:**Este documento representa la identificación de Stakeholders y sus roles a partir de la interpretación de uso del sistema.
3. **Actividades de arquitectura en el ciclo de vida:**  
   N/A
4. **Usos de las descripciones de arquitectura:**Las descripciones de arquitectura de este documento se usarán para referenciar el diseño del sistema de software iAccess.

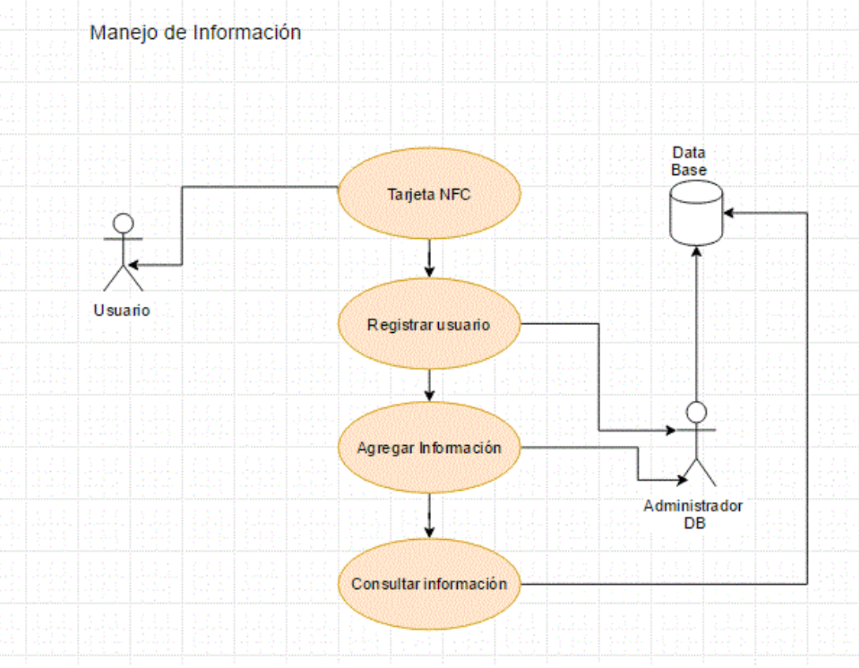
**5. Descripciones prácticas de arquitectura:**

**5.1 Documentación de la arquitectura:** N/A.  
  
**5.2 Identificación de los Stakeholders y sus responsabilidades:** EJEMPLO.

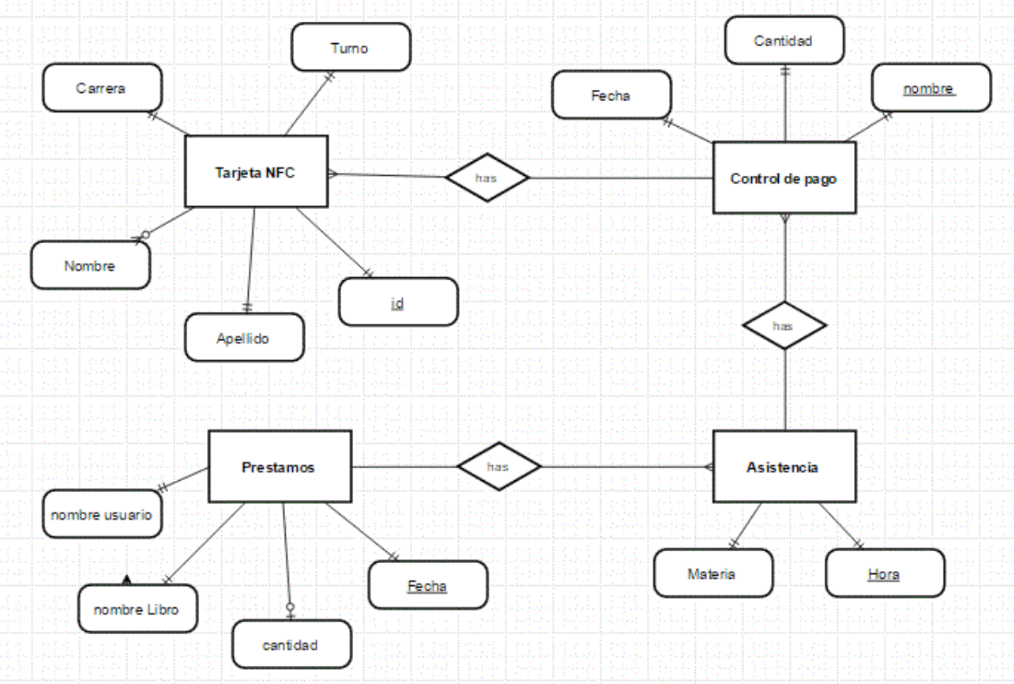
|  |  |  |  |
| --- | --- | --- | --- |
| **Stakeholder** | **Descripción** | **Escenario** | **Vistas** |
| Administrador. | -El administrador gestionará nuevos alumnos y actualizará datos. | -Registro de usuario.  -Modificación de datos.  -Eliminación de alumnos. | -CU manejo de información. |
| Alumno | -La persona que interactúa con el sistema y acude al  administrador  para la  administración  de datos. | -Registro de usuarios.  -Modificación de datos.  -Entrada al plantel.  -Préstamo de libros. | -CU Control de pagos.  -CU Asistencia.  -CU Manejo de  información  -Préstamo de libros. |

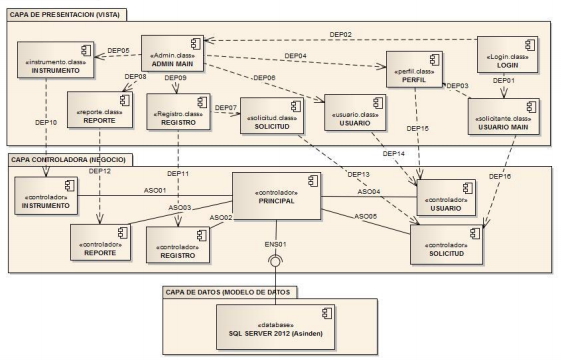
**5.3 Selección de puntos de vista de la arquitectura:** EJEMPLO.

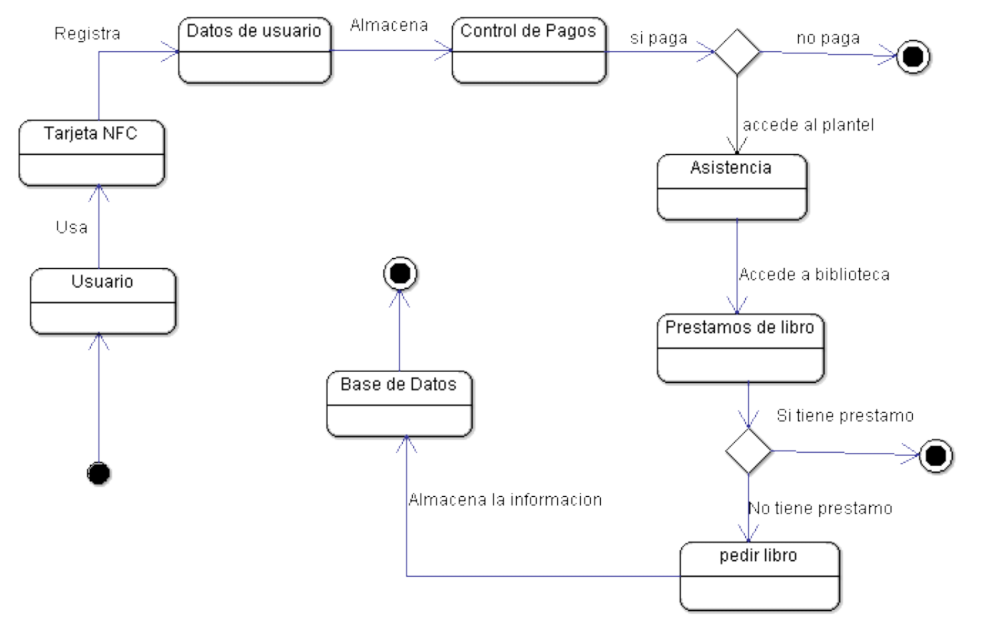
|  |  |
| --- | --- |
| **Vistas** | **UML** |
| Escenarios | Casos de uso |
| Lógica | Clases |
| Desarrollo | Componentes |
| Fisica | Estados |
| Procesos | Secuencia |

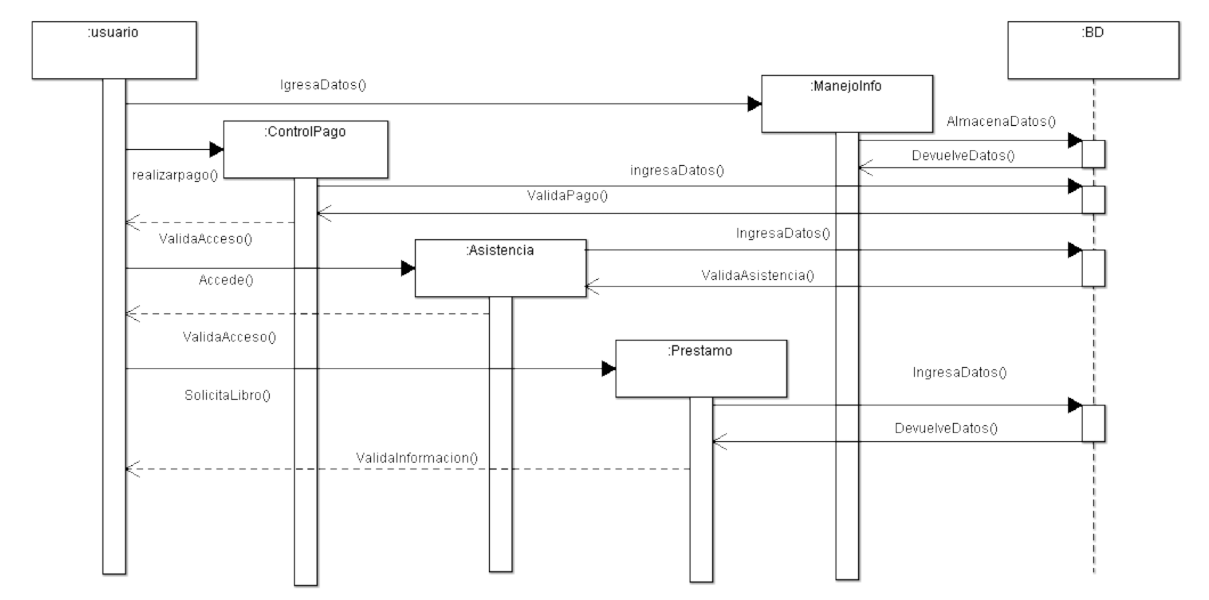
**5.4 Vistas de la arquitectura:** Vistas - Escenario: EJEMPLO.Diagrama de casos de uso: Manejo de información.  
  


**Vista - Lógica:** (diagramas de clase preferible) EJEMPLO.  
Diagrama entidad relación.

****

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


**Vista - Física:**  EJEMPLO.Diagrama de estados.  


**Vista - Procesos:**  EJEMPLO.Diagrama de secuencia.  


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

**Descripcion de modulos.** EJEMPLO:

|  |  |  |
| --- | --- | --- |
| **Nombre del módulo** | **Descripción** | **Componentes incluidos** |
| Capa De Presentación  (vista) | Módulo que contiene toda la parte visual y perceptible para el usuario final. Estas vistas son los formularios de JFrame que ofrecen interfaces para aplicaciones de escritorio  desarrolladas en Java. | - LOGIN  - PERFIL  - USUARIO MAIN  - ADMIN MAIN  - INSTRUMENTO  - REGISTRO  - REPORTE  - SOLICITUD  - USUARIO |
| Capa de Negocio (Controlador) | Este módulo es el encargado de  realizar la comunicación entre el modelo y las vistas, es quien  controla la lógica entre peticiones de las vistas para dirigirlas a su respectivo modelo de datos. | - INSTRUMENTO  - REGISTRO  - REPORTE  - PRINCIPAL  - SOLICITUD  - USUARIO |
| Capa de datos  (MODELO) | Módulo que contiene las clases que  hacen persistencia a la base de  datos, de tal forma que puedan  generar y traer información de la  base de datos. | - INSTRUMENTO  - SOLICITUD  - LISTA\_SOLICITUD  - REGISTRO  - LISTA\_REGISTRO  - USUARIO |

**Descripción de componentes.** *“Se describen los componentes del ‘Diagrama de componentes’”. EJEMPLO.  
  
En la columna “COMPONENTES RELACIONADOS” de la siguiente tabla, cada componente tendrá*

*una de las siguientes siglas correspondiente al paquete que lo contiene:*

*- (V): Capa de presentación (VISTA).*

*- (C): Capa de negocio (CONTROLADOR).*

*- (M): Capa de datos (MODELO).*

*Además, los tipos de relaciones que se aprecian en el diagrama de componentes son:*

*- [As]: Asociación.*

*- [De]: Dependencia.*

*- [En]: Ensamblar.*

|  |  |  |
| --- | --- | --- |
| ***Componente*** | ***Descripción*** | ***Componentes relacionados*** |
|  | *Vista que permite la*  *autenticación de usuarios (tanto encargado como unidad dental) para ingresar al aplicativo.* | *- ADMIN MAIN (V)[De]*  *- USUARIO MAIN (V)[De]* |
|  | *Vista que tiene como objetivo entregar una interfaz para editar el perfil de un usuario del*  *aplicativo.* | *- USUARIO (C)[De]*  *- ADMIN MAIN (V)[De]*  *- USUARIO MAIN (V)[De]* |
|  | *Vista principal o el home para la*  *unidad dental (usuario), contiene la lógica para solicitar*  *instrumentos y ver el perfil de usuario.* | *- PERFIL (V)[De]*  *- SOLICITUD (C)[De]*  *- LOGIN (V)[De]* |

**Descripción de conectores.** *“Se describen los conectores del diagrama de componentes”. EJEMPLO.*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Nombre conector** | **Tipo componente** | **Relaciona** | | **Descripción de contenido** |
| **DEP01** | Vista | LOGIN | USUARIO MAIN | El usuario UNIDAD DENTAL  se autentifica en la vista  LOGIN, con el propósito de  ingresar a la vista de  USUARIO MAIN (home del  usuario unidad dental). |
| **DEP02** | Vista | LOGIN | ADMIN MAIN | El usuario ENCARGADO se  autentifica en la vista  LOGIN, con el propósito de  ingresar a la vista de  USUARIO ADMIN (HOME  del usuario encargado). |
| **DEP03** | Vista | USUARIO MAIN | PERFIL | El usuario UNIDAD DENTAL  puede ingresar a la vista de  PERFIL si desea editar su  perfil. |

**5.6 Arquitectura lógica:**

**Performances:** EJEMPLO.  
  
1) El sistema tendrá tiempos de respuesta menores a 5 segundos en cada transacción con la base de datos del servidor local.

2) El sistema será capaz de soportar el acceso simultáneo de más de 10 usuarios.

3) El aplicativo requerirá de un espacio en el disco duro menor a 100Mb y uso no mayor a 32Mb de RAM.

4) La lectura de instrumentos dentales no toma más de 10 segundos en realizarla para un registro.

5) El sistema trabajará bajo un entorno de red local en el mismo establecimiento.

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:** (Se listan los atributos de calidad y se describen respecto al proyecto). EJEMPLO.

|  |  |  |
| --- | --- | --- |
| **Atributo de calidad** | **Descripción** | **Métrica** |
| **Confiabilidad** | El sistema debe estar operativo al  menos un 99% durante un año. | Acceso a información  controlada por parámetros de seguridad. |
| **Usabilidad** | El sistema debe ser capaz de realizar  todas las transacciones de manera  sencilla e intuitiva. | Niveles de profundidad menor a 4 desde un menú principal, para lograr hacer una función  del aplicativo. Encuesta de satisfacción respecto al uso del aplicativo. |

**5.7 Ejemplo de uso:**

N/A.

**5.8 Detalles de la implementación:**

**5.8.1 Lenguajes y plataformas:**Para la implementación del sistema “NOMBRE” se requieren los siguientes lenguajes y plataformas:

1- Entorno de ejecución Windows como SO.

2- Servidor de base de datos SQL SERVER (2012 recomendable).

3- Un entorno con tecnología RFID (Lectores y Etiquetas RFID).

4- Tener actualizado JAVA en la versión 45 en adelante.