**Instituto Tecnológico de Culiacán**

**Ingeniería en Sistemas Computacionales**

**Arquitectura de Software**

**Dra. María Lucia Barrón Estrada**

**Arquitectura de Software del "Sistema Integral para Administración de Condominios"**

**Integrantes:**

**Barrón Zambada Juan Antonio**

**Sandoval López Alejandro**

**Sosa Toscano Luis Lorenzo**

**26/Mayo/2014**

Índice

Ingeniería en Sistemas Computacionales

Arquitectura de Software

Dra. María Lucia Barrón Estrada

Enero-junio 2014

**Primera parte del proyecto semestral**

**Fecha de entrega: 28 abril 2014**

Usando la descripción del proyecto “Sistema Integral para Administración de Condominios” analiza la información y obtén lo siguiente:

1. Identifica el conjunto de actores
2. Un diagrama de contexto para el sistema
3. Conjunto de Arquetipos y sus relaciones
4. Describe tres instancias (de la vida real) en donde podría ser aplicado el sistema
5. Clasifica los requisitos funcionales y de calidad y asígnales prioridades.
6. Identifica los escenarios más importantes y desarrolla casos de uso para cada uno de ellos. (Nota, en este momento los escenarios tal vez no estarán completamente desarrollados, recuerda que los requisitos pueden cambiar.)
7. Usando los casos de uso, identifica los componentes necesarios para crear un modelo inicial para este dominio y encuentra las relaciones entre ellos para presentar la estructura del sistema.
8. Incluye las tablas de mapeo de los casos de uso a los componentes. De los requisitos a los casos de uso.
9. Ten presentes los atributos de calidad pero no es necesario que estos se reflejen en forma explícita en el análisis.

Documenta tu trabajo completamente, incluye un índice de los temas que se van a presentar. Describe el problema y organiza la solución que planteas lo más detalladamente posible.

**Descripción del Proyecto**

**“Sistema Integral para Administración de Condominios” (SIAC)**

El proyecto consiste en diseñar, evaluar y transformar la arquitectura de software de un sistema para administración de condominios. El sistema ayudara al administrador de un fraccionamiento urbano a llevar un control más preciso a cerca de las entradas y salidas de dinero así como mantener la información actualizada.

Los requerimientos de usuario son los siguientes:

1. Existe una persona que se encarga de administrar el fraccionamiento.
2. Existe un grupo de condóminos los cuales pueden ser propietarios de una o más casas o departamentos.
3. El administrador puede conocer el monto de dinero con el que se cuenta en cualquier fecha.
4. Los condóminos (personas que viven en el fraccionamiento) pagan una cuota mensual de mantenimiento.
5. Las cuotas de mantenimiento se establecen de acuerdo al tamaño del terreno que ocupe la propiedad.
6. El administrador emite recibos por los pagos realizados por los condóminos.
7. Los recibos de pagos están foliados.
8. El administrador envía recordatorios de pago mensualmente a aquellos condóminos que no están al corriente de sus pagos.
9. Los condóminos pueden realizar sus pagos física o electrónicamente.
10. El administrador paga a los trabajadores su sueldo semanalmente. Recabando la firma de cada uno de ellos en una nómina semanal.
11. Existen cuatro tipos de trabajadores (Limpieza, Guardia, Mantenimiento, Administrador)
12. El administrador verifica la asistencia de los trabajadores cada día.
13. El sueldo de los trabajadores se calcula en base a los días trabajados.
14. El administrador paga periódicamente los servicios del fraccionamiento (agua, luz, teléfono, alberca, etc.) Estos se registran en los gastos y se archivan los recibos oficiales.
15. El administrador informa mensualmente a los condóminos a cerca de las entradas y salidas de dinero, así como de la información que le es requerida (lista de condóminos con deudas, gastos extras no contemplados, etc.)
16. Los condóminos pueden acceder sus estados de cuenta en línea.
17. El administrador compra todos los utensilios y materiales necesarios para proporcionar los servicios de limpieza y mantenimiento de áreas comunes. Registrándolos en el sistema y almacenando los recibos correspondientes.
18. Las puertas de acceso al fraccionamiento son controladas por un sistema de seguridad y alarmas.
19. El guardia registra las entadas y salidas al fraccionamiento de automóviles y personas.
20. Las áreas de uso común (alberca, salones, canchas, etc. ) deben ser solicitadas al guardia o administrador para su uso en eventos especiales.
21. Las visitas deben estar registradas en el sistema.
22. Los condóminos pueden obtener reportes de estados financieros generales del fraccionamiento o reportes estadísticos de gastos por nomina, servicios, materiales, etc.
23. Los usuarios del sistema deberán identificarse para tener acceso al mismo.

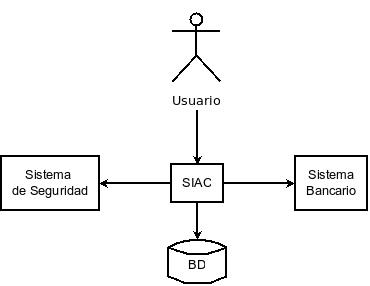
**Lista de Actores**

**Actores**

|  |  |  |
| --- | --- | --- |
| **Actores** | **Objetivo** | |
| **Administrador** | Es el usuario principal que está en uso directo con el sistema, encargado de las entradas y salidas de datos del mismo | |
| **Condóminos** | Es el dueño del condominio a administrar por lo tanto es el principal interesado en lo reportes que el sistema genere. | |
| **Trabajadores** | Se registran en el sistema para llevar con control de asistencia al trabajo. | |
| **Sistema de Seguridad** | | Lleva en control de las personas y vehículos que accedan al fraccionamiento. Tendrá acceso a la base de datos de sistema para llevar el control de acceso de entradas y salidas. |
| **Banco** | | Se comunica con el sistema para los pagos que se realizan de manera electrónica. |

**Diagrama de contexto**

Se ha estructurado el diagrama de contexto desacuerdo a la información que hemos recabado sobre los requerimientos dados por el cliente y la descripción del problema. Aquí se muestra el Diagrama de contexto.

****

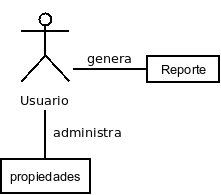
**Los arquetipos y sus relaciones:**

**Usuarios:** Este será la persona que utilizará el sistema.

**Los diferentes tipos de Usuarios son:**

* Administradores
* Condóminos
* Trabajadores

**Propiedades:** son los terrenos, edificios o lugares que se vana administrar mediante el sistema.

****Los diferentes tipos de propiedades son:

* Casas
* Terrenos
* Departamentos

**Reportes:** Es una abstracción de los diferentes tipos de reportes que el sistema puede generar

.

**Requisitos Funcionales y de Calidad**

**Requisitos Funcionales**

**RF1**.- El sistema debe tener un Administrador

**RF2**.- Los condóminos pueden registrar más de una vivienda a su nombre

**RF3**.- El administrador tendrá acceso a lo que el tesorero hace.

**RF4**.- El sistema controlara los pagos del mantenimiento de los condóminos y se podrán realizar en efectivo o de forma electrónica

**RF5**.- Se calculara la cuota mensual automáticamente a partir del tamaño de la vivienda,(Configurable al tamaño del terreno).

**RF6**.- El sistema deberá llevar un control de los pagos y emitirá recibos de estos mismos foliados.

**RF7**.- El sistema generara avisos a los condóminos morosos (vía web).

**RF8**.- Un subsistema informara al sistema sobre la salida de dinero por pago a trabajadores.

**RF9**.- Llevar un control de pago a los trabajadores dependiendo del área en que laboran (4 áreas) limpieza, guardia, Admón., etc.

**RF10**.- Administrador debe verificar asistencia.

**RF11**.- Sueldo calculado en base a la asistencia

**RF12**.- Llevara un control de salida por servicios (agua, luz, teléfono, etc.) y se registraran los gastos y se archivaran los recibos.

**RF13**.- El sistema llevara un registro de las entradas y salidas de dinero, creara un reporte mensual.

**RF14**.- El sistema dará acceso a los condóminos a un reporte acerca de sus estados de cuenta a través de internet.

**RF15**.- El sistema dará acceso a su base de datos al sistema de seguridad y alarma para el control del acceso al condominio.

**RF16**.- El sistema llevara un registro de tráfico de personas y vehículos que accedan al fraccionamiento

**RF17**.- El sistema llevara un registro de los permisos para el uso de las áreas de uso común.

**RF18**.- El sistema llevara un registro de visitas externas a los condóminos

**RF19**.- Al final de cada periodo el sistema generara un reporte del estado financiero y de nomina del fraccionamiento

**RF20**.- El sistema tendrá una interfaz de acceso a los condóminos para identificarse como propietario y así tener acceso al fraccionamiento

**RF21**.- El sistema debe poder realizar altas, bajas y cambios de condóminos y de empleados.

**Requisitos de Calidad.**

**RC1**- El sistema debe ser seguro: El sistema de encriptación deberá usar un sistema de encriptación para la información.

**RC2**.- El sistema debe ser confiable: El sistema debe estar disponible las 24 horas.

**RC3**.- El sistema debe ser fácil de usar: La interfaz del sistema debe ser fácil de entender.

**RC4**.- El sistema debe ser rápido: El sistema debe responder a las peticiones de los usuarios en menos de 5 segundos.

**RC5**.- El sistema debe ser de forma eficiente: El debe realizar sus tareas de forma correcta.

**Prioridades**

|  |  |  |
| --- | --- | --- |
| **Requisito** | **Prioridad** | **Validación** |
| **RF1** | Alta | Es necesario que exista el administrador para poder utilizar el sistema. |
| **RF2** | Alta | Es necesario que los Condóminos puedan tener más de 1 condominio a su nombre. |
| **RF3** | Alta | Es necesario dado a que el interesado en los datos financieros es el administrador. |
| **RF4** | Alta | Es necesario para que condóminos conozcan su estado cuenta y puedan pagarlo de la forma que más les convenga. |
| **RF5** | Alta | Es necesario para calcular los pagos de las viviendas. |
| **RF6** | Alta | Es necesario debido a que se debe llevar un registro de los pagos que se están haciendo. |
| **RF7** | Alta | Se debe llevar un control de los condóminos morosos para llevar una buena administración. |
| **RF8** | Baja | El sistema deberá proveer este servicio más adelante ya que no es realmente relevante al sistema. |
| **RF9** | Baja | El sistema deberá proveer este servicio más adelante ya que no es realmente relevante al sistema. |
| **RF10** | Baja | El sistema deberá proveer este servicio más adelante ya que no es realmente relevante al sistema. |
| **RF11** | Baja | El sistema debe clasificar las salidas de dinero por los pagos de servicios para llevar una buena administración financiera. |
| **RF12** | Alta | Es necesario que el sistema cree reportes para ayudar al administrador en las tomas de decisiones respecto a .la administración de los condominios. |
| **RF13** | Media | Es necesario que los condóminos conozcan el estado financiero de sus condominios para saber sus adeudos y cuando deben pagarlos. |
| **RF14** | Media | El sistema de seguridad es un sistema externo así que no es de mucha prioridad. |
| **RF15** | Media | El sistema debe cumplir este requisito para ayudar al sistema de seguridad a controlar el acceso al fraccionamiento. |
| **RF16** | Alta | El sistema deberá llevar control de personas debido a que es una petición del cliente por cuestiones de seguridad. |
| **RF17** | Media | Es necesario llevar un control de las áreas comunes para evitar problemas a la hora de agendar su uso, no es un requisito alto debido a que el sistema aun así puede trabajar sin él. |
| **RF18** | Media | No es indispensable para la administración de los condominios pero es necesario para la seguridad. |
| **RF19** | Alta | Los reportes son importantes para llevar la administración de los condominios. |
| **RF20** | Alta | Es necesaria para poder comunicarse con el sistema y trabajar con el. |
| **RF21** | Alta | Es importante poder realizar cambios a los usuarios del sistema para la administración de los condominios. |
| **RC1** | Alta | El sistema debe asegurar que la información que maneja solo debe ser accedida y utilizada por las personas pertinentes. |
| **RC2** | Alta | El acceso a la plataforma debe ser garantizado 24/7 |
| **RC3** | Alta | El sistema debe ser fácil de usar para los usuarios. |
| **RC4** | Media | El tiempo de respuesta a una petición no debe ser mayora a 2 segundos |
| **RC5** | Alta | El sistema debe asegurar que la información mostrada es correcta |

**Escenarios**

Estos son algunos escenarios que podrían presentarse en el uso del “Sistema Integral para Administración de Condominios” (**SIAC**).

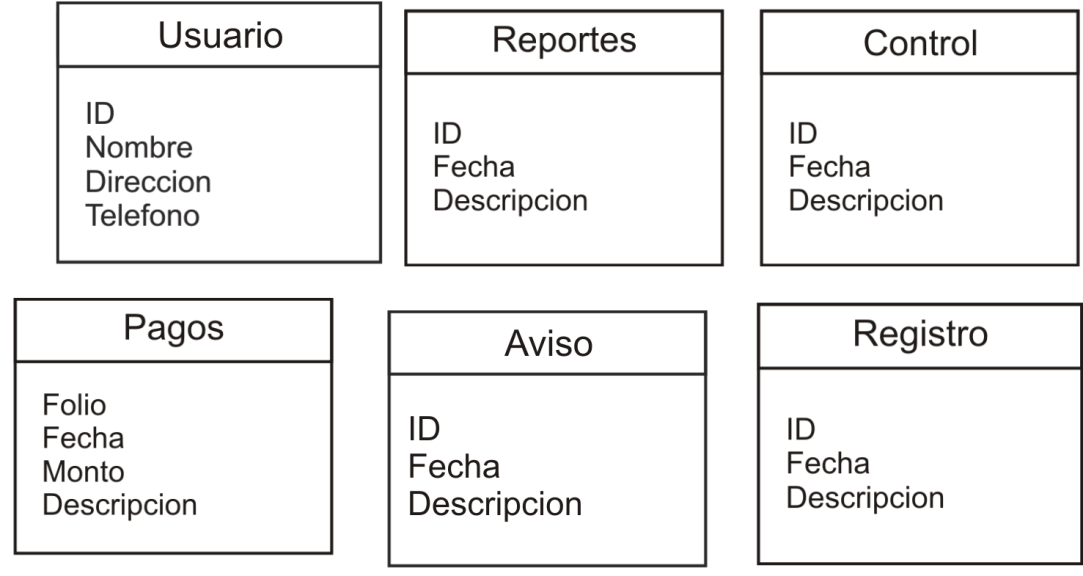
|  |  |
| --- | --- |
| **Escenario #1** | **Registro y modificación de condóminos** |
| El administrador accede al SIAC para registrar un nuevo condómino o modificar la información alguno ya existente. | |
| **Escenario #2** | **Acceso al SIAC** |
| El usuario ingresa al SIAC, proporciona su cuenta de usuario y contraseña correcta para acceder a la información pertinente a su rol (administrador, condómino o trabajador). | |
| **Escenario #3** | **El SIAC genera el informe mensual financiero** |
| El administrador accede al SIAC para obtener el informe financiero mensual del fraccionamiento y el sistema se lo muestra según los parámetros que el administrador selecciono. | |
| **Escenario #4** | **Pago de cuotas** |
| Los condóminos acceden al SIAC para conocer sus cuotas y adeudos, además el sistema les presenta las formas de pago (transferencia bancaria, deposito, efectivo, cheque, etc.) | |

**Componentes:**

Basados en los Casos de Uso anteriormente definidos se pudo identificar el siguiente conjunto de componentes:

* C1. Usuarios
* C2 Reportes
* C3. Control
* C4. Pagos
* C5. Avisos
* C6. Registro
* C7 Web

**Clases conceptuales.** Las clases que se identificaron fueron:



**Casos de Uso**

A continuación mostramos el listado de los Casos de Uso identificados.

|  |  |
| --- | --- |
| **CU #** | **Nombre** |
| **CU01**.- | Ingresar al Sistema |
| **CU02**.- | Asignar Viviendas a Condóminos |
| **CU03**.- | Pago de Servicios |
| **CU04**.- | Generar Avisos |
| **CU05**.- | Verificar Asistencia |
| **CU06**.- | Sueldo |
| **CU07**.- | Control de Salida de Dinero |
| **CU08**.- | Registro de Dinero |
| **CU09**.- | Crear Reporte |
| **CU10**.- | Revisión de Estado de Cuenta |
| **CU11**.- | Control de Personas |
| **CU12**.- | Control de Visitas |
| **CU13**.- | Generar Reportes |
| **CU14**.- | Identificarse |
| **CU15**.- | Altas, Bajas y Cambios |
| **CU16**.- | Control de Áreas Comunes |

La siguiente tabla muestra las descripciones de los Casos de Uso.

|  |  |  |
| --- | --- | --- |
| **Caso de Uso** | **Nombre del Caso de Uso** | **Descripción del CU** |
| **CU01** | Ingresar al Sistema | El usuario se identifica ante el sistema |
| **CU02** | Asignar Viviendas a Condóminos | El usuario asignara una o más viviendas a los condóminos |
| **CU03** | Pago de Servicios | El usuario podrá realizar sus pagos de servicios a través del sistema |
| **CU04** | Generar Avisos | El sistema generara avisos para los Condóminos morosos |
| **CU05** | Verificar Asistencia | El usuario administrador deberá verificar la asistencia de sus trabajadores |
| **CU06** | Sueldo | El sueldo se calculara a través de la asistencia |
| **CU07** | Control de Salida de Dinero | El sistema llevara un control de salida de dinero por Servicios y generara recibos |
| **CU08** | Registro de Dinero | El sistema llevara un registro de las entradas y salidas de dinero |
| **CU09** | Crear Reporte mensual | El sistema creara y mostrara un reporte mensual de los registros de dinero |
| **CU10** | Revisión de estado de cuenta | El usuario podrá visualizar su estado de cuenta actual en el sistema. |
| **CU11** | Control de Personas | El sistema confirmara los permisos de las personas para entrar y salir del lugar |
| **CU12** | Control de Visitas | El sistema llevara un control de los visitantes a los condominios. |
| **CU13** | Generar Reportes Mensuales | El usuario podrá generar reportes mensuales financieros. |
| **CU14** | Identificarse | El usuario se identificara como propietario de un condominio para tener acceso al mismo. |
| **CU15** | Altas, Bajas y Cambios | El usuario podrá dar de alta, baja o hacer cambios, sobre los condominios y de empleados |
| **CU15** | Control de Áreas Comunes |  |

**Tabla de Mapeo**

La siguiente es la tabla de mapeo de los requisitos con respecto a los Casos de Uso.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Tabla de mapeo** | | | | | | | | | | | | | | | |
|  | **Casos de Uso** | | | | | | | | | | | | | | |
| **RF** | **CU**  **01** | **CU**  **02** | **CU**  **03** | **CU**  **04** | **CU**  **05** | **CU**  **06** | **CU**  **07** | **CU**  **08** | **CU**  **09** | **CU**  **10** | **CU**  **11** | **CU**  **12** | **CU**  **13** | **CU**  **14** | **CU**  **15** |
| **RF01** | **x** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **RF02** |  | **x** |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **RF03** |  | **x** | **x** | **x** | **x** |  |  |  |  |  |  |  |  |  |  |
| **RF04** |  |  | **x** |  |  |  |  |  |  |  |  |  |  |  |  |
| **RF05** |  |  | **x** |  |  |  |  | **x** |  |  |  |  |  |  |  |
| **RF06** |  |  |  |  |  |  | **x** | **x** |  |  |  |  |  |  |  |
| **RF07** |  |  |  |  |  |  | **x** |  |  |  |  |  |  |  |  |
| **RF08** |  |  |  | **x** |  |  |  |  |  |  |  |  |  |  |  |
| **RF09** |  |  |  | **x** |  |  |  |  |  |  |  |  |  |  |  |
| **RF10** |  |  |  | **x** |  |  |  |  |  |  |  |  |  |  |  |
| **RF11** |  |  |  |  |  |  |  | **x** | **x** |  |  |  | **x** |  |  |
| **RF12** |  |  |  |  |  |  |  |  | **x** |  |  |  | **x** |  |  |
| **RF13** |  |  | **x** | **x** |  |  |  |  | **x** | **x** |  |  |  |  | **x** |
| **RF14** |  |  |  |  |  |  |  |  |  |  |  |  |  | **x** |  |
| **RF15** |  |  |  |  |  |  |  |  |  |  | **x** | **x** |  |  |  |
| **RF16** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **RF17** |  |  |  | **x** |  |  | **x** | **x** | **x** |  |  |  |  |  |  |
| **RF18** |  |  |  |  |  |  |  |  |  | **x** |  |  |  |  |  |
| **RF 19** |  |  |  |  |  |  |  |  | **x** |  |  |  | **X** |  |  |
| **RF 20** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **RF 21** |  |  |  |  |  |  |  |  |  |  |  |  |  |  | **x** |
| **RF** | **CU**  **01** | **CU**  **02** | **CU**  **03** | **CU**  **04** | **CU**  **05** | **CU**  **06** | **CU**  **07** | **CU**  **08** | **CU**  **09** | **CU**  **10** | **CU**  **11** | **CU**  **12** | **CU**  **13** | **CU**  **14** | **CU**  **15** |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Componentes | Casos de Uso | | | | | | | | | | | | | | | | | |
| CU  1 | CU  2 | CU  3 | CU  4 | CU  5 | CU  6 | | CU  7 | CU  8 | CU  9 | CU  10 | CU  11 | | CU  12 | CU  13 | | CU  14 | CU  15 |
| C1 | X | X | x |  |  |  | |  |  |  | X |  | |  | X | | X | X |
| C2 |  |  |  |  |  |  | | X |  | X |  |  | |  | X | |  |  |
| C3 |  | X |  |  | X |  | | X |  |  | X | X | | X |  | |  | X |
| C4 |  |  | X |  |  | X | |  | X | X |  |  | |  | X | |  |  |
| C5 |  |  |  | X |  |  | |  |  |  |  |  | |  |  | |  |  |
| C6 |  |  |  |  |  |  | |  | X | X |  |  | | X |  | | X | X |
| C7 |  |  | X |  |  | |  |  |  | X |  | |  |  |  |  | |  |

Mapeo de Componentes frente a casos de uso

**Posibles modelos Arquitectónicos**

Los posibles modelos que podría tener el sistemas de SISA “Sistema de Alfabetización” y se presentaran a continuación.

|  |  |  |  |
| --- | --- | --- | --- |
| **Posibles Modelos Arquitectónicos** | | | |
| **Tipo de modelo arquitectónico** |  | | |  |
| **Ventajas** | **Desventajas** | |
| Filtros y Mangueras | * Su simplicidad permite al diseñador comprender más fácil las entradas y salidas del sistema. * Permite la concurrencia ya que cada filtro puede trabajar por sí solo. * Fácil mantenimiento y mejora. | * No hay cooperación entre filtros. * Tantos filtros pueden transformar mucho la información y crear problemas en el desempeño. * Mientras los filtros procesan la información el usuario casi no tiene interacción con el sistema. | |
| La forma en que trabaja esta arquitectura limita mucho la interacción entre el usuario y el sistema. | | |
| Orientada a objetos | * Permite a los diseñadores descomponer problemas en agentes que interactúan entre sí. * Mas intuitivo ya que usamos objetos del mundo real en la implementación del software. * Facilidad para distribuir objetos. * Como un objeto oculta su representación a sus clientes, es posible cambiar su implementación sin modificar los clientes. | * Para que un objeto interactúe con otro (mediante la invocación a un procedimiento) debe conocer la identidad del otro objeto. Luego, cuando la identidad de un objeto cambie es necesario modificar todas las invocaciones a tal objeto. * Se pueden presentar efectos laterales: si los objetos A y C usan al objeto B, entonces los efectos de C en B lucen como efectos laterales no esperados en A, y viceversa. | |
| Este modelo nos proporciona una alta reutilización al igual que proporciona separación por módulos lo que nos proporciona un mejor acoplamiento. | | |
| Capas | * Soportan diseño basado en niveles incrementales de abstracción. * Facilidad de reusabilidad y de mejoramiento. | * No todos los sistemas pueden fácilmente ser diseñados basados en capas. * Usar muchas capas aumenta el riesgo de error. | |
| Ayuda casi a una totalidad de los requisitos, existirá una mayor interacción y más rápida comunicación entre usuarios, los componentes por capas pueden ser reutilizables. | | |
| Pizarra | * Puede guardar grandes cantidades de información de forma eficiente * Gestión centralizada. | * Dificultad Para distribuir la información * Evolución de la información es cara. | |
| Es una arquitectura interactiva con el usuario, pero está más dada para un número mayor de subsistemas diferentes integrados en uno solo. | | |
| Eventos | * Mucha Reusabilidad. * Facilidad para mejorar el Sistema. * Componentes pueden ser cambiados por otros sin afectar los demás componentes del sistema. | * Cuando un componente no tiene idea como otros componentes responderán a este. * El desempeño y el manejo de recursos pueden llegar a ser un problema serio. | |
| Es una arquitectura muy mantenible, más rápido en el manejo de eventos, su desventaja es que al haber demasiada concurrencia fallara. | | |
| Cliente-servidor | * Centralización del control de los recursos, datos y accesos. * Facilidad de mantenimiento y actualización del lado del servidor: Esto es porque el lado del servidor se puede mantener o actualizar fácilmente. Por ejemplo, una actualización se aplica a un único servidor, pero los beneficios los obtienen múltiples clientes generalmente sin necesidad de que éstos actualicen nada. * Toda la información es almacenada en el lado del servidor, que suele tener mayor seguridad que los clientes. * Hay muchas herramientas cliente-servidor probadas, seguras y amigables para usar. | | * Si el número de clientes simultáneos es elevado, el servidor puede saturarse. Esto sucede con menor frecuencia en las redes P2P. * Frente a fallas del lado del servidor, el servicio queda paralizado para los clientes. Algo que no sucede en una red P2P. |
|  | Es una arquitectura distribuida que permite a los usuarios finales obtener acceso a la información en forma transparente aún en entornos multiplataforma | | |
| Servicios | * Reduce el nivel de acoplamiento. * Clara definición de roles de desarrollo. * Definición de seguridad más clara. * Fácil testeo. * Mejora la mantención. * Favorece la reutilización. * Favorece el desarrollo en paralelo. * Permite fácil escalabilidad. * Permite un mapeo directo entre los procesos y los sistemas. * Permite un monitoreo preciso. * Permite la interoperabilidad. | | * SOA depende de la implementación de estándares. Sin estándares, la comunicación entre aplicaciones requiere de mucho tiempo y código. * SOA no es para: aplicaciones con alto nivel de transferencia de datos, aplicaciones que no requieren de implementación del tipo request/response y para aplicaciones que tienen un corto periodo de vida. * Incrementalmente se hace difícil y costoso el ser capaz de cumplir con los protocolos y hablar con un servicio. * Implica conocer los procesos del negocio, clasificarlos, extraer las funciones que son comunes a ellos, estandarizarlas y formar con ellas capas de servicios que serán requeridas por cualquier proceso de negocio. |
| Es una manera de diseñar e implementar los procesos de negocios, obteniendo una gran flexibilidad en su mantención y evolución. | | |

# Arquitectura Propuesta

Llegamos a la conclusión que la arquitectura que mejor se adapta a las necesidades para el desarrollo del sistema es la arquitectura por capas ya que por soporta diseño basado en niveles incrementales de abstracción, facilidad de reusabilidad y de mejoramiento; lo que proporciona un mayor tiempo de respuesta y por su reusabilidad hace que el mantenimiento del sistema sea muy sencillo.

Mapeo de Componentes