**Documento de Arquitectura (SAD)**

**Casa Inteligente**



**Realizado por:**

|  |  |  |
| --- | --- | --- |
| **Persona** | **Rol** | **Código Uniandes** |
| Carlos Ernesto González Vargas | Ingeniero de Requerimientos | 200819123 |
| Sandra Milena Gómez Ríos | Ingeniero de Producción | 201110951 |
| Andrés Mauricio Erazo Benavides | Ingeniero de Soporte | 201110949 |
| David Pérez Chibuque | Ingeniero de Calidad | 201117818 |
| Willian Alejandro Idrobo Luna | Arquitecto Jefe | 201110544 |
| Erik Fernando Arcos Franco | Ingeniero de Administración | 201110856 |

**Control de versiones**

|  |  |  |  |
| --- | --- | --- | --- |
| **Versión** | **Fecha** | **Autor** | **Descripción del Cambio** |
| 1.00 | 4 de Junio | Ingenium | Creación del documento |
| 1.01 | 8 de Junio | Ingenium | Revisión final del documento |
|  |  |  |  |
|  |  |  |  |

 

**Contenido**

**Pag.**

[**1.** **Descripción del Documento** 6](#_Toc298086855)

[**1.1.** **Propósito y Audiencia** 6](#_Toc298086856)

[**1.2.** **Organización del Documento** 6](#_Toc298086857)

[**1.3.** **Terminología y Definiciones** 6](#_Toc298086858)

[**1.4.** **Documentos Relevantes** 6](#_Toc298086859)

[**2.** **Generalidades del Proyecto** 7](#_Toc298086860)

[**2.1.** **Problema a Resolver** 7](#_Toc298086861)

[**2.2.** **Descripción General del Sistema a Desarrollar** 7](#_Toc298086862)

[**2.3.** **Objetivos de la Arquitectura** 7](#_Toc298086863)

[**2.4.** **Roles del Equipo de Trabajo** 8](#_Toc298086864)

[**2.5.** **Stakeholders** 9](#_Toc298086865)

[**3.** **Motivadores Arquitecturales** 10](#_Toc298086866)

[**3.1.** **Motivadores de Negocio** 10](#_Toc298086867)

[**3.2.** **Restricciones** 11](#_Toc298086868)

[**3.2.1.** **Restricciones Técnicas** 11](#_Toc298086869)

[**3.2.2.** **Restricciones de Negocio** 12](#_Toc298086870)

[**3.3.** **Atributos de Calidad** 12](#_Toc298086871)

[**3.3.1.** **Escenarios de Calidad** 12](#_Toc298086872)

[**3.3.2.** **árbol de Utilidad** 16](#_Toc298086873)

[**4.** **Contexto** 18](#_Toc298086874)

[**4.1.** **Escenarios Operacionales** 18](#_Toc298086875)

[**5.** **Puntos de Vista y Modelos Arquitecturales** 21](#_Toc298086876)

[**5.1.** **Punto de Vista Funcional** 21](#_Toc298086877)

[**5.1.1.** **Descripción** 21](#_Toc298086878)

[**5.1.2.** **Modelo de Componentes** 21](#_Toc298086879)

[**5.2.** **Punto de Vista de Despliegue** 22](#_Toc298086880)

[**5.2.1.** **Descripción** 22](#_Toc298086881)

[**5.2.2.** **Modelos de Plataforma de Ejecución** 22](#_Toc298086882)

[**5.2.3.** **Modelos de Red** 24](#_Toc298086883)

[**5.2.4.** **Modelos de Dependencia Tecnológica** 25](#_Toc298086884)

[**5.3.** **Punto de Vista de Información** 25](#_Toc298086885)

[**5.3.1.** **Descripción** 25](#_Toc298086886)

[**5.3.2.** **Modelos de Estructuras Estáticas de Datos** 25](#_Toc298086887)

[**5.3.3.** **Modelos de Flujo de Información** 27](#_Toc298086888)

[**5.3.4.** **Modelos de Ciclo de Vida de Información** 30](#_Toc298086889)

[**5.4.** **Punto de Vista de Concurrencia** 31](#_Toc298086890)

[**5.4.1.** **Descripción** 31](#_Toc298086891)

[**5.4.2.** **Modelos** 31](#_Toc298086892)

[**5.5.** **Punto de Vista de Desarrollo** 31](#_Toc298086893)

[**5.5.1.** **Descripción** 31](#_Toc298086894)

[**5.5.2.** **Modelos** 32](#_Toc298086895)

[**6.** **Relaciones entre los Puntos de Vista** 35](#_Toc298086896)

[**7.** **Evaluación de la Arquitectura** 36](#_Toc298086897)

[**7.1.** **Atam** 36](#_Toc298086898)

[**7.2.** **Experimentación** 36](#_Toc298086899)

[**7.2.1.** **Experimento 1 - Desempeño** 36](#_Toc298086900)

[**7.2.1.1.** **Experimento A – Latencia** 36](#_Toc298086901)

[**7.2.1.1.1.** **A1 – Registro de Alarma** 36](#_Toc298086902)

[**7.2.1.1.2.** **A2 – Consulta de Información Consolidada** 38](#_Toc298086903)

[**7.2.1.2.** **Experimento B – Escalabilidad** 41](#_Toc298086904)

[**7.2.1.2.1.** **B1 – Escalabilidad del Sistema** 41](#_Toc298086905)

[**7.2.2.** **Experimento 2 – Seguridad y Disponibilidad** 43](#_Toc298086906)

[**7.2.2.1.** **Experimento A – Seguridad** 43](#_Toc298086907)

[**7.2.2.1.1.** **A1 – Autenticación y Autorización** 44](#_Toc298086908)

[**7.2.2.1.2.** **A2 – Denegación de Servicio / Suplantación** 46](#_Toc298086909)

[**7.2.2.2.** **Experimento B – Disponibilidad** 48](#_Toc298086910)

[**7.2.2.2.1.** **B – Disponibilidad** 48](#_Toc298086911)

[**7.2.3.** **Experimento 3 - Interoperabilidad** 50](#_Toc298086912)

[**7.2.3.1.** **Experimento A - Comunicación con Web Service** 50](#_Toc298086913)

[**7.2.3.2.** **Experimento B - Comunicación Asincrona – Envío de correo electrónico** 55](#_Toc298086914)

[**7.2.3.3.** **Experimento C - Video llamada** 55](#_Toc298086915)

[**8.** **Bibliografía** 56](#_Toc298086916)

**Índice de Tablas**

**Pag.**

[Tabla 1. Roles ACDM 8](#_Toc298086917)

[Tabla 2. Listado de Stakeholders 9](#_Toc298086918)

[Tabla 3. Stakeholders y Expectativas 9](#_Toc298086919)

[Tabla 4. Motivador de Negocio M1 10](#_Toc298086920)

[Tabla 5. Motivador de Negocio M2 10](#_Toc298086921)

[Tabla 6. Motivador de Negocio M3 11](#_Toc298086922)

[Tabla 7. Restricción Técnica RT01 11](#_Toc298086923)

[Tabla 8. Restricción Técnica RT02 11](#_Toc298086924)

[Tabla 9. Restricción Técnica RT03 12](#_Toc298086925)

[Tabla 10. Restricción de Negocio RN01 12](#_Toc298086926)

[Tabla 11. Restricción de Negocio RN02 12](#_Toc298086927)

[Tabla 12. Escenario de Calidad EC01 12](#_Toc298086928)

[Tabla 13. Escenario de Calidad EC02 13](#_Toc298086929)

[Tabla 14. Escenario de Calidad EC03 13](#_Toc298086930)

[Tabla 15. Escenario de Calidad EC04 13](#_Toc298086931)

[Tabla 16. Escenario de Calidad EC05 14](#_Toc298086932)

[Tabla 17. Escenario de Calidad EC06 14](#_Toc298086933)

[Tabla 18. Escenario de Calidad EC07 14](#_Toc298086934)

[Tabla 19. Escenario de Calidad EC08 15](#_Toc298086935)

[Tabla 20. Escenario de Calidad EC09 15](#_Toc298086936)

[Tabla 21. Escenario de Calidad EC10 15](#_Toc298086937)

[Tabla 22. Árbol de Utilidad 16](#_Toc298086938)

[Tabla 23. Escenario operacional EO01 18](#_Toc298086939)

[Tabla 24. Escenario operacional EO02 18](#_Toc298086940)

[Tabla 25. Escenario operacional EO03 19](#_Toc298086941)

[Tabla 26. Escenario operacional EO04 19](#_Toc298086942)

[Tabla 27. Escenario operacional EO05 20](#_Toc298086943)

[Tabla 28. Escenario operacional EO06 20](#_Toc298086944)

[Tabla 29. Modelo de Componentes 21](#_Toc298086945)

[Tabla 30. Catálogo de Módulos 22](#_Toc298086946)

[Tabla 31. Modelo de Plataforma de Ejecución 23](#_Toc298086947)

[Tabla 32. Modelo de Red 24](#_Toc298086948)

[Tabla 33. Modelo de dependencia tecnológica 25](#_Toc298086949)

[Tabla 34. Modelo de estructura estática de datos 26](#_Toc298086950)

[Tabla 35. Descripción de entidades 26](#_Toc298086951)

[Tabla 36. Modelo de flujo de información: Evento 28](#_Toc298086952)

[Tabla 37. Modelo de flujo de información: Regla item 29](#_Toc298086953)

[Tabla 38. Modelo de ciclo de vida de información 30](#_Toc298086954)

[Tabla 39. Modelo de concurrencia del sistema 31](#_Toc298086955)

[Tabla 40. Modelo de desarrollo del sistema 32](#_Toc298086956)

[Tabla 41. Descripción del Experimento 1 - A1 36](#_Toc298086957)

[Tabla 42. Resultados del Experimento 1 - A1 37](#_Toc298086958)

[Tabla 43. Descripción del Experimento 1 - A2\_1 38](#_Toc298086959)

[Tabla 44. Resultados del Experimento 1 - A2\_1 38](#_Toc298086960)

[Tabla 45. Descripción del Experimento 1 - A2\_2 39](#_Toc298086961)

[Tabla 46. Resultados del Experimento 1 - A2\_2 40](#_Toc298086962)

[Tabla 47. Descripción del Experimento 1 - B1 41](#_Toc298086963)

[Tabla 48. Resultados del Experimento 1 - B1 42](#_Toc298086964)

[Tabla 49. Descripción del Experimento 2 - A1 44](#_Toc298086965)

[Tabla 50. Resultados del Experimento 2 - A1 45](#_Toc298086966)

[Tabla 51. Descripción del Experimento A2 46](#_Toc298086967)

[Tabla 52. Resultados del Experimento 2 - A2 47](#_Toc298086968)

[Tabla 53. Descripción del Experimento 2 - B 48](#_Toc298086969)

[Tabla 54. Resultados del Experimento 2 - B1 49](#_Toc298086970)

[Tabla 55. Descripción del Experimento 3 - A 51](#_Toc298086971)

[Tabla 56. Resultados del Experimento 3 -A1 51](#_Toc298086972)

**Documento de Arquitectura (SAD)**

**Casa Inteligente**

1. **Descripción del Documento**
   1. **Propósito y Audiencia**

El Documento de Arquitectura del Sistema presenta la arquitectura de software propuesta para el proyecto ***Casa Inteligente*** planteado por la empresa **AlpesSmartHome (ASH)**. Se pretende por medio de este documento brindarles a los miembros de la empresa **AlpesSmartHome (ASH)** interesados, una visión global y comprensible del diseño a alto nivel para el proyecto ***Casa Inteligente***.

* 1. **Organización del Documento**

El documento se organiza y desarrolla de la siguiente manera:

* **Sección 1:** Presenta una descripción del documento en cuanto al propósito, público objetivo y terminología empleada.
* **Sección 2:** Presenta las generalidades del proyecto donde se describe el problema a resolver, la propuesta de solución a desarrollar, los objetivos establecidos y los stakeholders involucrados.
* **Sección 3:** Describe los motivadores de negocio, restricciones y atributos de calidad (junto con sus respectivos escenarios de calidad y árbol de utilidad) considerados en el proyecto ***Casa Inteligente***.
* **Sección 4:** Presenta los escenarios operacionales considerados.
  1. **Terminología y Definiciones**
* **SAD:** Documento de Arquitectura del Sistema
* **ASH:** Empresa AlpesSmartHome
* **RFID:** Siglas de **Radio FrequencyIdentification** o Identificación por radiofrecuencia.
  1. **Documentos Relevantes**

A continuación se muestra la lista de documentos relevantes que se han tenido en cuenta para la preparación de este documento:

* **Proyecto – Casa Inteligente, Uniandes, 2011.** Presenta la definición base del proyecto propuesto.
* **Documento de Arquitectura del Sistema (SAD), Uniandes, 2011.** Sirve como plantilla para la creación de este documento.

1. **Generalidades del Proyecto**
   1. **Problema a Resolver**

La empresa **Alpes Smart Home (ASH)** desea entrar a participar en el mercado de las casas inteligentes dado el considerable avance de la tecnología en dispositivos móviles, sensores inalámbricos y sistemas embebidos que ha habido durante los últimos años. Por esta razón la empresa **ASH** plantea el proyecto ***Casa Inteligente***, el cual surge para dar respuesta al diseño de arquitectura de solución de un sistema que permita la administración de viviendas inteligentes.

* 1. **Descripción General del Sistema a Desarrollar**

El proyecto ***Casa Inteligente*** tiene las siguientes características básicas definidas por la empresa **ASH**:

* El sistema debe permitir controlar oficinas o viviendas ofreciendo acceso remoto e información de estado a los diferentes propietarios y administradores.
* El sistema central recopila la información que los sensores envían al sistema con la información de estado. La información recibida es procesada de forma que cada propietario pueda acceder desde su cuenta al sistema para administrar y consultar el estado de su vivienda u oficina.
* Los administradores podrán gestionar la información del sistema central de inmuebles y usuarios autorizados de cada una de las viviendas u oficinas que se encuentre bajo su control.
* El sistema debe permitir que cada urbanización o conjunto de oficinas aumente el número de viviendas bajo su control
* Las operaciones críticas deben ser atendidas en manera inmediata.
* Los propietarios de una vivienda pueden consultar la información del sistema desde dispositivos móviles o clientes web de forma que resulte intuitiva y ágil.
* Cada electrodoméstico dentro de una casa, puede ser marcado con un tag RFID de forma que se sepa e informe al sistema en todo momento donde se encuentra cada activo registrado en la vivienda.
* Un usuario puede configurar alarmas para determinar las zonas y horarios autorizados para un determinado activo.
* El propietario de una vivienda puede tener reportes de actividades de su vivienda y consultar todos los eventos sucedidos en un determinado periodo de tiempo, igualmente los administradores debe poder generar reportes de forma instantánea a partir de la información disponible en el sistema.
  1. **Objetivos de la Arquitectura**

A continuación se presentan los objetivos:

* Proporcionar elementos que colaboren con la toma de decisiones para el desarrollo del proyecto.
* Aportar conceptos y un lenguaje común que permitan la comunicación entre los equipos que participen en un proyecto.
* Diseñar una arquitectura que permita cumplir con los atributos de calidad que soporten la operación del sistema.
* Elaborar un diseño breve y conciso con mayor cohesión y bajo acoplamiento posible entre cada módulo.
  1. **Roles del Equipo de Trabajo**

Con base en la experiencia que el grupo de trabajo de Ingenium ha tenido en su trabajo en equipo, y la experiencia profesional de sus integrantes, se llegó a la conclusión que cada uno de los integrantes llevará a cabo los siguientes roles durante la ejecución del proyecto:

Tabla 1. Roles ACDM

|  |  |  |
| --- | --- | --- |
| **Rol** | **Descripción** | **Responsable** |
| **Ingeniero de Administración** | Responsable de coordinar el diseño del sistema. Puede ser el gerente del proyecto en algunas ocasiones. Debe coordinar, planear y dirigir todas las actividades de diseño. | Erik Fernando  Arcos Franco |
| **Ingeniero de Soporte** | Responsable por instalar y configurar las herramientas de diseño, tales como los ambientes de desarrollo, las herramientas de configuración, los ambientes y herramientas de pruebas, etc. Participa en el diseño del sistema desde el punto de vista de infraestructura. | Andrés Mauricio  Erazo Benavides |
| **Arquitecto Jefe** | Responsable del diseño completo del sistema. Trabaja con los demás miembros del equipo para construir el diseño del sistema, comenzando por los motivadores de la arquitectura, diseño, revisión y evaluación de la arquitectura hasta llevarla a producción. Responsable de la creación y mantenimiento del documento de arquitectura. | Willian Alejandro  Idrobo Luna |
| **Ingeniero de Requerimientos** | Participan en la documentación de los motivadores de la arquitectura, así como el control de cambios sobre los mismos. Este rol sirve como punto de contacto con el cliente. Participa junto con el ingeniero de calidad en la revisión de la arquitectura y durante las pruebas de la arquitectura. | Carlos Ernesto  González Vargas |
| **Jefe científico** | Este rol es responsable por coordinar, planear, seguir y documentar los experimentos necesarios para refinar el diseño de la arquitectura. Se concentra en aspectos técnicos que pueden complicar la arquitectura. Asiste al ingeniero de calidad en las revisiones de diseño. | Este Rol va a ser llevado por todo el equipo de desarrollo |
| **Ingeniero de calidad** | Este rol se encarga de garantizar que los procesos sean seguidos de acuerdo a lo estipulado. Se encarga de coordinar las actividades de revisión de diseño, así como de coordinar las pruebas. Durante las revisiones, este rol debe capturar, documentar y seguir los problemas encontrados en el diseño. | David Pérez  Chibuque |
| **Ingeniero de Producción** | Este rol se encarga del diseño detallado, la implementación de los elementos arquitecturales, y la integración del sistema. | Sandra Milena  Gómez Ríos |

La selección de Roles se realizó por medio de una encuesta publicada en el siguiente link:

[**https://spreadsheets.google.com/spreadsheet/viewform?hl=en\_US&formkey=dENEZWNsalhneGhJdVN4OHUtZk1ENVE6MQ**](https://spreadsheets.google.com/spreadsheet/viewform?hl=en_US&formkey=dENEZWNsalhneGhJdVN4OHUtZk1ENVE6MQ)

* 1. **Stakeholders**

Tabla 2. Listado de Stakeholders

|  |  |
| --- | --- |
| **Stakeholder** | **Descripción** |
| **Gerente** | Miembro de la empresa **Alpes Smart Home (ASH)**. Gerente encargado de tomar las decisiones de negocio y maneja todo el tema financiero (presupuesto y aprobaciones) para el proyecto ***Casa Inteligente*** |
| **Project Manager** | Miembro de la empresa **Alpes Smart Home (ASH)** y gerente del proyecto ***Casa Inteligente***. Encargado de licitar y seleccionar la arquitectura de software que satisfaga las necesidades del proyecto. |
| **Director Departamento IT** | Miembro de la empresa **Alpes Smart Home (ASH).**Dirige el departamento de tecnología de la empresa y será el encargado de administrar la solución tecnológica desarrollada. |
| **Comerciales** | Empleados de la empresa **AlpesSmartHome (ASH)** que conforman el equipo comercial de la empresa. Son los encargados de vender los inmuebles al cliente (propietarios) ofreciendo las ventajas del sistema. |
| **Propietario** | Propietarios de las viviendas. Son los compradores y/o clientes de la empresa **Alpes Smart Home (ASH)** y más específicamente, los usuarios finales del sistema a desarrollar. |

Tabla 3. Stakeholders y Expectativas

|  |  |
| --- | --- |
| **Stakeholder** | **Expectativas** |
| **Gerente** | El proyecto se realice dentro del costo, tiempo y alcance planeado.  Con el desarrollo del sistema se incrementen las ventas de los inmuebles. |
| **Project Manager** | El tiempo de implementación del proyecto sea de 4 meses.  El alcance y el costo se cumplan de acuerdo a lo planeado. |
| **Director Departamento IT** | El sistema sea seguro y no ponga en riesgo ni la seguridad del mismo sistema, ni la de la de los actores del mismo.  El sistema se pueda recuperar rápidamente ante un fallo o caída.  El sistema tenga alta disponibilidad y sea escalable.  El sistema sea fácil de administrar. |
| **Comerciales** | Incrementar sus promedios de ventas. |
| **Propietario** | El sistema cumpla con estándares de usabilidad (fácil de usar, fácil de administrar).  El sistema sea confiable y seguro. |

1. **Motivadores Arquitecturales**
   1. **Motivadores de Negocio**

Tabla 4. Motivador de Negocio M1

|  |  |  |
| --- | --- | --- |
| **Nombre** | **Descripción** | |
| **Incursionar en el mercado viviendas inteligentes.** | Obtener el 10% de las ventas que se realizan el mercado de las viviendas inteligentes en Colombia mediante la implantación de un sistema de manejo y control para las viviendas. | |
| **Medida del Impacto** | | |
| Obtención del mercado de viviendas inteligentes medido en porcentaje | | |
| **Rangos** | **Cota Mínima** | **Cota Máxima** |
| **Ninguno** | 0% | 2.9% |
| **Bajo** | 3% | 3.9% |
| **Moderado** | 4% | 5.9% |
| **Fuerte** | 6% | 7.9% |
| **Muy Fuerte** | 8% | >10% |
| **Asociación del Motivador con el Negocio** | **Definido Por:** |  |
| **Ejecutado Por:** |  |
| **Ubicación en el Portafolio del negocio** |  |

Tabla 5. Motivador de Negocio M2

|  |  |  |
| --- | --- | --- |
| **Nombre** | **Descripción** | |
| **Incrementar ventas por la construcción de viviendas** | Incrementar el 15% de las ventas de viviendas mediante la construcción de viviendas inteligentes. | |
| **Medida del Impacto** | | |
| Crecimiento de las ventas en la venta de viviendas inteligentes medido en millones de pesos | | |
| **Rangos** | **Cota Mínima** | **Cota Máxima** |
| **Ninguno** | 0 millones | 0.9 millones |
| **Bajo** | 1 millón | 99 millones |
| **Moderado** | 100 millones | 499 millones |
| **Fuerte** | 500 millones | 899 millones |
| **Muy Fuerte** | 900 millones | > 900 millones |
| **Asociación del Motivador con el Negocio** | **Definido Por:** |  |
| **Ejecutado Por:** |  |
| **Ubicación en el Portafolio del negocio** |  |

Tabla 6. Motivador de Negocio M3

|  |  |  |
| --- | --- | --- |
| **Nombre** | **Descripción** | |
| **Disminuir costos en la construcción de viviendas inteligentes** | Disminuir los costos de construcción en un 15% mediante el uso de tecnologías estándar libres. | |
| **Medida del Impacto** | | |
| Reducción de costos de construcción medido en porcentaje | | |
| **Rangos** | **Cota Mínima** | **Cota Máxima** |
| **Ninguno** | 0 | 1% |
| **Bajo** | 1% | 3% |
| **Moderado** | 3% | 6% |
| **Fuerte** | 6% | 10% |
| **Muy Fuerte** | 10% | >15% |
| **Asociación del Motivador con el Negocio** | **Definido Por:** |  |
| **Ejecutado Por:** |  |
| **Ubicación en el Portafolio del negocio** |  |

* 1. **Restricciones**
     1. **Restricciones Técnicas**

Tabla 7. Restricción Técnica RT01

|  |  |  |
| --- | --- | --- |
| **ID Restricción:**  **RT01** | **Tipo:**  **Tecnología** | **Nombre:**  **Solución basada en Web** |
| **Descripción:** | La solución debe ser Web para permitir la minimización de costos de mantenimiento y administración del sistema. | |
| **Establecida por:** | Director Departamento IT | |
| **Alternativas:** | No se proporcionan alternativas | |
| **Observaciones:** | Las aplicaciones web son populares debido a lo práctico del navegador web como cliente ligero, a la independencia del sistema operativo, así como a la facilidad para actualizar y mantener aplicaciones web sin distribuir e instalar software a miles de usuarios potenciales. | |

Tabla 8. Restricción Técnica RT02

|  |  |  |
| --- | --- | --- |
| **ID Restricción:**  **RT02** | **Tipo:**  **Tecnología** | **Nombre:**  **Plataformas Tecnológicas** |
| **Descripción:** | El sistema debe ser desarrollado principalmente bajo JEE y otras plataformas libres para permitir minimizar los costos de mantenimiento y la administración del sistema. | |
| **Establecida por:** | Director Departamento IT | |
| **Alternativas:** | No se proporcionan alternativas | |
| **Observaciones:** | Uno de los beneficios de Java EE como plataforma es que es posible empezar con poco o ningún coste. La implementación Java EE puede ser descargada gratuitamente, y hay muchas herramientas de código abierto, disponibles para extender la plataforma o para simplificar el desarrollo. | |

Tabla 9. Restricción Técnica RT03

|  |  |  |
| --- | --- | --- |
| **ID Restricción:**  **RT03** | **Tipo:**  **Tecnología** | **Nombre:**  **Acceso Móvil** |
| **Descripción:** | El sistema debe ser accesible por medio de una interfaz especializada para dispositivos móviles | |
| **Establecida por:** | Director Departamento IT | |
| **Alternativas:** | Desarrollar una aplicación por plataforma: Android, Rim, Java | |
| **Observaciones:** | El acceso a una página web por medio de un dispositivo móvil no es lo mismo que el acceso a la misma página por medio de un browser, el sistema debe ser capaz de detectar el navegador con el cual fue accedido y configurar el layout de tal forma que sea adecuado para la pantalla reducida de un dispositivo móvil. | |

* + 1. **Restricciones de Negocio**

Tabla 10. Restricción de Negocio RN01

|  |  |  |
| --- | --- | --- |
| **ID Restricción:**  **RN01** | **Tipo:**  **Negocio** | **Nombre:**  **Limitación de Presupuesto** |
| **Descripción:** | Se podrá disponer de un equipo humano máximo de 6 personas. | |
| **Establecida por:** | Project Manager y Gerente | |
| **Alternativas:** | No se proporcionan alternativas | |
| **Observaciones:** | Eltiempoypresupuestoparaeldesarrollosolopermitepagarunequipodemáximo6personasdurante4mesesparala implementación del producto. | |

Tabla 11. Restricción de Negocio RN02

|  |  |  |
| --- | --- | --- |
| **ID Restricción:**  **RN02** | **Tipo:**  **Negocio** | **Nombre:**  **Tiempo al mercado / Limitaciones de Agenda** |
| **Descripción:** | El proyecto debe ser realizado en 4 meses. | |
| **Establecida por:** | Project Manager y Gerente | |
| **Alternativas:** | No se proporcionan alternativas | |
| **Observaciones:** | Silaarquitecturanoestálistaen4mesessepuedellegaraperderunnegociovitalparalasubsistencia de la empresa. | |

* 1. **Atributos de Calidad**
     1. **Escenarios de Calidad**

Tabla 12. Escenario de Calidad EC01

| **Escenario de Calidad #** | **EC01** | **Stakeholder:** | **Propietario, Project Manager** |
| --- | --- | --- | --- |
| **Atributo de Calidad** | Desempeño | | |
| **Justificación** | El sistema debe ser capaz de alertar en menos de un segundo a las autoridades competentes en caso de que se presente una emergencia, en este caso, un incendio | | |
| **Fuente** | Sensor | | |
| **Estímulo** | Mensaje de alerta de humo | | |
| **Artefacto** | Sistema | | |
| **Ambiente** | Sistema en estado normal, líneas telefónicas disponibles | | |
| **Respuesta** | Lanzamiento de llamada telefónica, el sistema continua en estado normal | | |
| **Medida de la Respuesta** | El lanzamiento de la llamada es en menos de un segundo | | |

Tabla 13. Escenario de Calidad EC02

| **Escenario de Calidad #** | **EC02** | **Stakeholder:** | **Director Departamento TI** |
| --- | --- | --- | --- |
| **Atributo de Calidad** | Desempeño | | |
| **Justificación** | El sistema debe mantener su nivel de servicio hasta unas determinadas condiciones de carga del sistema, de manera que todas las peticiones de los clientes sean respondidas dentro de un tiempo determinado. | | |
| **Fuente** | Usuario externo | | |
| **Estímulo** | 100 consultas a la vez | | |
| **Artefacto** | Sistema | | |
| **Ambiente** | Recursos del sistema consumidos en un 80% | | |
| **Respuesta** | El sistema sigue en funcionamiento y dio respuesta a todas las solicitudes. | | |
| **Medida de la Respuesta** | Jitter menor igual a 2 segundos | | |

Tabla 14. Escenario de Calidad EC03

| **Escenario de Calidad #** | **EC03** | **Stakeholder:** | **Gerente, Project Manager, Propietario** |
| --- | --- | --- | --- |
| **Atributo de Calidad** | Seguridad | | |
| **Justificación** | La información del usuario debe ser confidencial y solo se debe poder consultar por el usuario o un administrador. | | |
| **Fuente** | Usuario externo: propietario | | |
| **Estímulo** | Consulta | | |
| **Artefacto** | Datos del sistema | | |
| **Ambiente** | El sistema se encuentra en funcionamiento normal, y más de 20 viviendas u oficinas registradas. | | |
| **Respuesta** | El sistema solo muestra al propietario la información de su vivienda u oficina. | | |
| **Medida de la Respuesta** | El 100% de las consultas deben retornar solo los datos de la vivienda oficina a la cual está asignado el usuario logueado. | | |

Tabla 15. Escenario de Calidad EC04

| **Escenario de Calidad #** | **EC04** | **Stakeholder:** | **Gerente, Project Manager, Director Departamento IT** |
| --- | --- | --- | --- |
| **Atributo de Calidad** | Modificabilidad | | |
| **Justificación** | El sistema debe ser escalable de manera que pueda soportar el crecimiento estimado por el gerente de **ASH** durante el futuro próximo. | | |
| **Fuente** | Usuario de la central | | |
| **Estímulo** | Agrega nuevas oficinas y/o viviendas hasta llegar a 900 viviendas u oficinas | | |
| **Artefacto** | Sistema | | |
| **Ambiente** | Condiciones normales de ejecución del sistema | | |
| **Respuesta** | El sistema no degrada sus servicios manteniendo los tiempos de respuesta de las acciones de los usuarios | | |
| **Medida de la Respuesta** | Se espera cumplir con este atributo agregando máximo 9 nuevos nodos al cluster donde se despliega el sistema, sin tener que realizar modificaciones a los componentes de software del sistema | | |

Tabla 16. Escenario de Calidad EC05

| **Escenario de Calidad #** | **EC05** | **Stakeholder:** | **Comercial, Propietario, Director departamento IT** |
| --- | --- | --- | --- |
| **Atributo de Calidad** | Disponibilidad | | |
| **Justificación** | La información del sistema debe estar a disposición del propietario cuando este la disponga con un nivel de al menos el 99% | | |
| **Fuente** | Usuario | | |
| **Estímulo** | Accede al sistema para consultar el estado de su vivienda u oficina | | |
| **Artefacto** | Sistema de monitoreo de viviendas | | |
| **Ambiente** | Fallo en un componente del sistema | | |
| **Respuesta** | El sistema debe permitir al usuario realizar las consultas básicas y críticas sobre el estado de la vivienda y sobre la localización de los bienes en ella | | |
| **Medida de la Respuesta** | El 100% de las consultas consideradas criticas deben seguir ejecutándose correctamente | | |

Tabla 17. Escenario de Calidad EC06

| **Escenario de Calidad #** | **EC06** | **Stakeholder:** | **Director Departamento IT** |
| --- | --- | --- | --- |
| **Atributo de Calidad** | Desempeño | | |
| **Justificación** | Los reportes del sistema deben ser accesibles por usuarios del tipo administrador. | | |
| **Fuente** | Usuario administrador | | |
| **Estímulo** | Generar reporte consolidado del estado de todas las viviendas | | |
| **Artefacto** | Sistema | | |
| **Ambiente** | Condiciones normales | | |
| **Respuesta** | El sistema genera un reporte mostrando usuarios y viviendas registradas en el sistema, actividades registradas sobre los electrodomésticos de las viviendas de una urbanización, número de alertas generadas por el sistema durante el día. | | |
| **Medida de la Respuesta** | El sistema tarda 2 segundos en generar y mostrar el reporte | | |

Tabla 18. Escenario de Calidad EC07

| **Escenario de Calidad #** | **EC07** | **Stakeholder:** | **Persona, Comercial, Project Manager** |
| --- | --- | --- | --- |
| **Atributo de Calidad** | Disponibilidad | | |
| **Justificación** | El propietario debe tener certeza de que va a ser alertado cuando haya alguna eventualidad referente a sus activos, por ejemplo que alguno de sus electrodomésticos sea retirado de su zona u horarios asignados. | | |
| **Fuente** | Persona | | |
| **Estímulo** | Extraer un electrodoméstico de un área en un horario no permitido | | |
| **Artefacto** | Sensores RFID | | |
| **Ambiente** | Condiciones normales | | |
| **Respuesta** | El sistema envía alerta al propietario de la vivienda indicando el evento que ocurrió | | |
| **Medida de la Respuesta** | El 99% de las veces que ocurre el estímulo bajo el ambiente especificado, el sistema responde correctamente | | |

Tabla 19. Escenario de Calidad EC08

| **Escenario de Calidad #** | **EC08** | **Stakeholder:** | **Project Manager** |
| --- | --- | --- | --- |
| **Atributo de Calidad** | Modificabilidad / Extensibilidad- Usabilidad | | |
| **Justificación** | El usuario debe poder acceder a la información de su vivienda y contar con las mismas funciones a través de las diversas plataformas establecidas: Browsers y Dispositivos móviles | | |
| **Fuente** | Propietario | | |
| **Estímulo** | Consultar la información del sistema | | |
| **Artefacto** | Sistema | | |
| **Ambiente** | Condiciones normales | | |
| **Respuesta** | Es posible consultar la misma información desde un Browser o un dispositivo móvil | | |
| **Medida de la Respuesta** | La información es consistente entre las 2 plataformas | | |

Tabla 20. Escenario de Calidad EC09

| **Escenario de Calidad #** | **EC09** | **Stakeholder:** | **Project Manager** |
| --- | --- | --- | --- |
| **Atributo de Calidad** | Desempeño | | |
| **Justificación** | Se debe tener una visión general del funcionamiento de los sensores en todo momento, de manera que aquellos que estén presentando un funcionamiento anormal, pe. Baja batería, sean cambiados a tiempo. | | |
| **Fuente** | Sensor | | |
| **Estímulo** | Mensaje de funcionamiento anormal | | |
| **Artefacto** | Sistema | | |
| **Ambiente** | Sistema en condiciones normales | | |
| **Respuesta** | Se registra en el sistema el error presentado y se envía una notificación al propietario y al administrador del sistema | | |
| **Medida de la Respuesta** | El sistema tarda 2 segundos en reportar el error | | |

Tabla 21. Escenario de Calidad EC10

| **Escenario de Calidad #** | **EC10** | **Stakeholder:** | **Propietario, Project Manager** |
| --- | --- | --- | --- |
| **Atributo de Calidad** | Desempeño | | |
| **Justificación** | Se debe poder dar una respuesta rápida ante el evento de un intruso de manera que se pueda actuar de manera correcta. | | |
| **Fuente** | Sensor | | |
| **Estímulo** | Mensaje de alerta de intruso | | |
| **Artefacto** | Sistema | | |
| **Ambiente** | Sistema en estado normal, líneas telefónicas disponibles | | |
| **Respuesta** | Lanzamiento de llamada telefónica, el sistema continua en estado normal | | |
| **Medida de la Respuesta** | El lanzamiento de la llamada es en menos de un segundo | | |

* + 1. **árbol de Utilidad**

Tabla 22. Árbol de Utilidad

| **Atributo** | **Medida** | **ID** | **Descripción** | **Prioridad Negocio** | **Prioridad Arquitecto** |
| --- | --- | --- | --- | --- | --- |
| **Desempeño** | **Jitter** | **EC02** | El sistema debe mantener su nivel de servicio hasta unas determinadas condiciones de carga del sistema, de manera que todas las peticiones de los clientes sean respondidas dentro de un tiempo determinado. | Normal | Alta |
| **Latencia** | **EC01** | El sistema debe ser capaz de alertar en menos de un segundo a las autoridades competentes en caso de que se presente una emergencia, en este caso, un incendio | Alta | Alta |
| **EC06** | Los reportes del sistema deben ser accesibles por usuarios del tipo administrador. | Normal | Normal |
| **EC09** | Se debe tener una visión general del funcionamiento de los sensores en todo momento, de manera que aquellos que estén presentando un funcionamiento anormal, pe. Baja batería, sean cambiados a tiempo. | Normal | Alta |
| **EC10** | Se debe poder dar una respuesta rápida ante el evento de un intruso de manera que se pueda actuar de manera correcta. | Alta | Alta |
| **Rendimiento** | **EC08** | El usuario debe poder acceder a la información de su vivienda y contar con las mismas funciones a través de las diversas plataformas establecidas: Browsers y Dispositivos móviles | Normal | Alta |
| **Disponibilidad** | **Tiempo de Disponibilidad** | **EC05** | La información del sistema debe estar a disposición del propietario cuando este la disponga con un nivel de al menos el 99% | Normal | Normal |
| **EC07** | El propietario debe tener certeza de que va a ser alertado cuando haya alguna eventualidad referente a sus activos, por ejemplo que alguno de sus electrodomésticos sea retirado de su zona u horarios asignados. | Alta | Normal |
| **Modificabilidad** | **Escalabilidad** | **EC04** | El sistema debe ser escalable de manera que pueda soportar el crecimiento estimado por el gerente de **ASH** durante el futuro próximo. | Normal | Alta |
| **Seguridad** | **Confidencialidad** | **EC03** | La información del usuario debe ser confidencial y solo se debe poder consultar por el usuario o un administrador. | Normal | Alta |

1. **Contexto**
   1. **Escenarios Operacionales**

Tabla 23. Escenario operacional EO01

|  |  |  |  |
| --- | --- | --- | --- |
| **Título del Escenario Operacional** | | | |
| Administración sistema central | | | |
| **Stakeholder Asociado** | Director Departamento IT | **ID** | EO01 |
| **Consideración Operacional** | **Respuesta del Stakeholder** | | |
| **Descripción general de la funcionalidad** | Se encarga de gestionar toda la información de inmuebles y usuarios registrados. | | |
| **¿Qué hace ahora el Stakeholder o le gustaría poder hacer?** | El administrador podrá ingresar, eliminar, modificar la información de inmuebles y los usuarios. | | |
| **Entradas provistas o disponible al momento del inicio** | El administrador debe tener disponible en el sistema toda la información de los activos y usuarios registrados, manual de instrucciones. | | |
| **Contexto de la operación** | Esta operación se realiza en el momento que se desea cambiar la información almacenada de los activos o usuarios. | | |
| **Respuesta del Sistema** | Al momento de ingresar los cambios al sistema, este deberá actualizar la información de forma que quede disponible a todos los usuarios. | | |
| **Salidas que el sistema produce como resultado de la acción** | La información o cambios realizados son vistos por los usuarios. | | |
| **¿Quién o qué usa la salida y para qué es utilizada?** | El sistema central utiliza las salidas para verificar y reportar la información que genera los activos a los diferentes usuarios autorizados. | | |

Tabla 24. Escenario operacional EO02

|  |  |  |  |
| --- | --- | --- | --- |
| **Título del Escenario Operacional** | | | |
| Generación de reportes | | | |
| **Stakeholder Asociado** | Director Departamento IT | **ID** | EO02 |
| **Consideración Operacional** | **Respuesta del Stakeholder** | | |
| **Descripción general de la funcionalidad** | Se encarga de generar reportes de las actividades u operaciones producidas durante un determinado tiempo. | | |
| **¿Qué hace ahora el Stakeholder o le gustaría poder hacer?** | El administrador podrá visualizar los reportes generados con la información disponible. | | |
| **Entradas provistas o disponible al momento del inicio** | Segmento de tiempo de interés para la generación del reporte. | | |
| **Contexto de la operación** | Esta operación se realiza cuando se debe obtener información instantánea de las actividades realizadas en las viviendas. | | |
| **Respuesta del Sistema** | El sistema debe entregar el reporte de actividades con la información disponible en él, de forma instantánea. | | |
| **Salidas que el sistema produce como resultado de la acción** | El sistema muestra la información solicitada en pantalla | | |
| **¿Quién o qué usa la salida y para qué es utilizada?** | Autoridades, Administrador del Sistema Central, las salidas es vista por tomar determinadas acciones de precauciones o alertas. | | |

Tabla 25. Escenario operacional EO03

|  |  |  |  |
| --- | --- | --- | --- |
| **Título del Escenario Operacional** | | | |
| Presentación de reportes y estado de vivienda | | | |
| **Stakeholder Asociado** | Propietarios | **ID** | EO03 |
| **Consideración Operacional** | **Respuesta del Stakeholder** | | |
| **Descripción general de la funcionalidad** | Se encarga de consultar y presentar la información de actividades y estado de los activos y sensores de la vivienda, así como de reportes de eventos sucedidos. | | |
| **¿Qué hace ahora el Stakeholder o le gustaría poder hacer?** | El propietario podría consultar y ver la información del estado de la vivienda cada vez que quiera y en un periodo de tiempo determinado. | | |
| **Entradas provistas o disponible al momento del inicio** | Datos del usuario para ingresar al sistema | | |
| **Contexto de la operación** | Esta operación se realiza cuando el propietario desea ingresar a su cuenta y revisar el estado de su vivienda. | | |
| **Respuesta del Sistema** | El sistema debe entregar el reporte correspondiente de actividades con la información disponible en él, de forma instantánea. | | |
| **Salidas que el sistema produce como resultado de la acción** | El sistema muestra la información solicitada en pantalla | | |
| **¿Quién o qué usa la salida y para qué es utilizada?** | Los propietarios lo emplean para verificar el estado de la vivienda actualmente o durante un periodo de tiempo determinado. | | |

Tabla 26. Escenario operacional EO04

|  |  |  |  |
| --- | --- | --- | --- |
| **Título del Escenario Operacional** | | | |
| Información de operación critica | | | |
| **Stakeholder Asociado** | Autoridades, Propietarios | **ID** | EO04 |
| **Consideración Operacional** | **Respuesta del Stakeholder** | | |
| **Descripción general de la funcionalidad** | Se encarga de informar cada vez que ocurra alguna operación crítica en la vivienda. | | |
| **¿Qué hace ahora el Stakeholder o le gustaría poder hacer?** | Recibir de forma instantánea el reporte de la operación crítica. | | |
| **Entradas provistas o disponible al momento del inicio** | Configuración del sistema con los dispositivos que reciben las alertas. | | |
| **Contexto de la operación** | Esta operación se ejecuta cuando algún sensor envía una alerta de una operación inusual o emergencia. | | |
| **Respuesta del Sistema** | El sistema debe enviar un mensaje de forma instantánea a las autoridades, propietario y administrador con la información de la operación. | | |
| **Salidas que el sistema produce como resultado de la acción** | El sistema muestra la información de la operación de la alerta en pantalla. | | |
| **¿Quién o qué usa la salida y para qué es utilizada?** | Administrador del sistema, autoridades, propietario: La información se utiliza para tomar alguna acción de contingencia. | | |

Tabla 27. Escenario operacional EO05

|  |  |  |  |
| --- | --- | --- | --- |
| **Título del Escenario Operacional** | | | |
| Recopilación de información del estado de los activos en vivienda | | | |
| **Stakeholder Asociado** | Propietarios | **ID** | EO05 |
| **Consideración Operacional** | **Respuesta del Stakeholder** | | |
| **Descripción general de la funcionalidad** | Se encarga de guardar la información que es enviada por los sensores instalados en la vivienda | | |
| **¿Qué hace ahora el Stakeholder o le gustaría poder hacer?** | Tener disponible registro de todas la actividades ocurridas en la vivienda | | |
| **Entradas provistas o disponible al momento del inicio** | Configuración de sensores | | |
| **Contexto de la operación** | Esta operación es realizada en funcionamiento normal todo el tiempo. | | |
| **Respuesta del Sistema** | El sistema debe recibir la información de los sensores y almacenarla en él. | | |
| **Salidas que el sistema produce como resultado de la acción** | El sistema reporta con éxito que la información se guardó. | | |
| **¿Quién o qué usa la salida y para qué es utilizada?** | El sistema la información guardada para generar reportes y presentar información a los usuarios. | | |

Tabla 28. Escenario operacional EO06

|  |  |  |  |
| --- | --- | --- | --- |
| **Título del Escenario Operacional** | | | |
| Reporte funcionamiento sensor | | | |
| **Stakeholder Asociado** | Propietario, Director Departamento IT | **ID** | EO06 |
| **Consideración Operacional** | **Respuesta del Stakeholder** | | |
| **Descripción general de la funcionalidad** | Los sensores informan al sistema cada determinado tiempo indicando que efectivamente están en funcionamiento. | | |
| **¿Qué hace ahora el Stakeholder o le gustaría poder hacer?** | Administrador: Recibir una notificación cada vez que un sensor bajo su administración deja de prestar el servicio.  Propietario: Recibir una notificación cada vez que un sensor de su vivienda deja de prestar servicio. | | |
| **Entradas provistas o disponible al momento del inicio** | Información y configuraciones de todos los sensores que están instalados y si están en funcionamiento. | | |
| **Contexto de la operación** | Esta operación se produce durante todo el tiempo que el sistema central esté en funcionamiento. | | |
| **Respuesta del Sistema** | Recibir respuesta de todos los sensores activos. | | |
| **Salidas que el sistema produce como resultado de la acción** | Enviar una notificación al momento de detectar que un sensor deja de funcionar. | | |
| **¿Quién o qué usa la salida y para qué es utilizada?** | Administradores del sistema, Propietarios: La salida se utiliza para verificar cual es el problema y tomar determinada acción para solucionarla. | | |

1. **Puntos de Vista y Modelos Arquitecturales**
   1. **Punto de Vista Funcional**
      1. **Descripción**

En este punto de vista se describen los elementos arquitecturales que proveen funcionalidad a los sistemas. Se especifican las relaciones entre ellos y sus respectivas responsabilidades.

* + 1. **Modelo de Descomposición**

El sistema AlpesSmartHomeSystem está compuesto de cuatro módulos de software principales relacionados entre ellos para darle funcionalidad al sistema completo. El modulo principal ASHCentralSystem está formado por módulos que se encargan de procesar información, reglas de negocio, realizar notificaciones y envío de información que es almacenada a una base de datos. ASHLoadBalancer se encarga de recibir información de los dispositivos externos para verificar su validez y realizar la entrega los ASHCentralSystem disponibles. ASHHomeSystem envía información del estado de la vivienda cada segundo. ASHWebModule provee la plataforma para generación de reporte, administración de usuarios.

Tabla 29. Modelo de Componentes

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | | | | | **Familia**  Module ( X )  C&C ( )  Allocation ( )  **Estilo arquitectural**  Capas  PACE  Coordinador |
| **Convención**  Relación |
| **Título:**  Modelo estático de descomposición | | **ID:**  PV01 | **Nivel Profundidad:**  Nivel 3 | | **Nomenclatura:**  Paquetes |
| **Arquitecto:**  Carlos González | **Grupo:**  INGENIUM | | | **Fecha:**  29/06/2011 | **Versión:**  1.0 |

Tabla 30. Catálogo de Módulos

|  |  |
| --- | --- |
| Modulo | Responsabilidades |
| ASHCentralSystem | Contiene toda la lógica de negocio para procesar los eventos recibidos por cada casa u oficina.  Administra la comunicación para notificar eventos a los sistemas externos.  Implementa una conexión a la base de datos donde se registran todos los eventos generados. |
| ASHLoadBalancer | Contiene la lógica encargada de re direccionar las peticiones recibidas a cada uno de los servidores centrales disponibles.  Implementa procesos encargados de manejar la seguridad del sistema, para evitar posibles ataques. |
| ASHHomeSystem | Contiene la lógica para el envío de los eventos generados por la casa u oficina. |
| ASHWebModule | Se encarga de la administración de los usuarios del sistema, creación y manejo de roles.  Consulta y genera informes para que los usuarios los visualicen. |

* 1. **Punto de Vista de Despliegue**
     1. **Descripción**

En este punto de vista describe el ambiente bajo el cual sistema va ser instalado y ejecutado. De esta forma se muestran sus nodos, relaciones entre ellos y forma de comunicación. Adicionalmente se presenta los requerimientos de red y dependencias tecnologías que presentan los diferentes sistemas en hardware y software en ambiente de ejecución.

* + 1. **Modelos de Plataforma de Ejecución**

En este modelo se presenta los nodos y módulos funcionales que hacen parte del sistema en ejecución.

Tabla 31. Modelo de Plataforma de Ejecución

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | | | | | **Familia**  Module ( )  C&C ( X )  Allocation ( )  **Estilo arquitectural**  Redundancia activa |
| **Convención**  Relación |
| **Título:**  Modelo de plataforma de ejecución | | **ID:**  PV01 | **Nivel Profundidad:**  Nivel 3 | | **Nomenclatura:**  UML |
| **Arquitecto:**  Mauricio Erazo | **Grupo:**  INGENIUM | | | **Fecha:**  29/06/2011 | **Versión:**  1.0 |

Como se puede observar, se emplean dos nodos Main Server, con el fin de ofrecer una alta disponibilidad a través del balaceo de cargar ofrecido por el Load Balancer Server el cual se encarga de repartir las peticiones que llegan por medio del Device Home hacia los nodos del Main Server. Estos nodos envían información valida al Data Server el cual almacena toda la información de los eventos enviados, a su vez el DataServer contiene envía la consulta de las reglas y configuraciones definidas para cada dispositivo hacia el componente del Main Server encargado de procesar estos datos.

El Main Server está encargado de comunicarse de forma directa por sockets a los sistemas externos en este caso al Police Server y EmergencySystem, para enviarles notificaciones de eventos o sucesos fuera de lo común que no cumplan las reglas definidas por los usuarios. Finalmente el Web Server se comunica con el Data Server para hacer las consultas necesarias para obtener los reportes de estado, además puede editar y consultar la información para la administración de usuarios y configuración sobre los diferentes dispositivos de la vivienda.

La comunicación entre servidores se realiza principalmente por sockets mediante el protocolo TCP, para la base de datos se utiliza el protocolo JDBC/ODBC.

* + 1. **Modelos de Red**

En este modelo se presenta los requerimientos de red necesarios en ambiente de ejecución para dar soporte a los requerimientos en la comunicación entre los sistemas.

Tabla 32. Modelo de Red

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | | | | | **Familia**  Module ( )  C&C ()  Allocation ( X)  **Estilo arquitectural**  Redundancia activa |
| **Convención**  Relación |
| **Título:**  Modelo de red | | **ID:**  PV01 | **Nivel Profundidad:** | | **Nomenclatura:**  RED |
| **Arquitecto:**  David Pérez | **Grupo:**  INGENIUM | | | **Fecha:**  29/06/2011 | **Versión:**  1.0 |

Para la comunicación entre servidores que hacen parte del sistema, es necesario tener disponible una red local con capacidad de transmisión de alta velocidad y soporte a gran cantidad de información, de modo que se favorezca un alto desempeño entre servidores del sistema. Para la comunicación de dispositivos externos en este caso sistemas externos como el Police Server y EmergencySystem Server se especifica para el Main Server posea tarjetas de red tipo Ethernet para la comunicación hacia sistemas externos.

* + 1. **Modelos de Dependencia Tecnológica**

En esta vista se definen las dependencias tecnológicas en ambiente de ejecución para hardware y software.

Tabla 33. Modelo de dependencia tecnológica

|  |  |
| --- | --- |
| Componente | Requiere |
| Servidor Aplicaciones (GlassFish 3.1) | Java JDK 6  JRE 6  Intel Core 2 Duo |
| Data Server | Windows 7 Home Premium  Oracle 10g XE  Intel Core 2 Duo |
| Central System | Windows 7 Home Premium  Java JDK 6  JRE 6  Intel Core 2 Duo |
| Main Server | Java JDK 6  JRE 6  Intel Core 2 Duo |
| Police Server | Windows 7 Home Premium |
| Emergency Server | Windows 7 Home Premium |
| Load Balancer | Windows 7 Home Premium  Java JDK 6  Intel Core 2 Duo |
| Device Home | PLC |
| Web Server | GlassFish 3.1  Ojdbc 14.jar  Intel Core 2 Duo |

* 1. **Punto de Vista de Información**
     1. **Descripción**

El punto de vista de información nos permitirá conocer en detalle la información que está relacionada con los procesos de negocio, el flujo dentro del sistema y los stakeholders involucrados.

* + 1. **Modelos de Estructuras Estáticas de Datos**

En este diagrama describimos el modelo entidad relación del sistema, sus entidades principales, interrelaciones y propiedades. Nos permite conocer en detalle los datos que requiere manejar el sistema y los datos necesarios para la estrategia de seguridad utilizada.

Tabla 34. Modelo de estructura estática de datos

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | | | | | **Familia**  Module ( X )  C&C ( )  Allocation ( )  **Estiloarquitectural**  PACE |
| **Convención**  Relación que indica que la cardinalidad es 0 o más.    Tabla de la base de datos que almacena información |
| **Título:**  Modelo de estructura de datos estática | | **ID:**  PV03 | **Nivel Profundidad:**  Nivel 1 | | **Nomenclatura:**  Entidad - relación |
| **Arquitecto:**  WillianIdrobo | **Grupo:**  INGENIUM | | | **Fecha:**  29/06/2011 | **Versión:**  1.0 |

Tabla 35. Descripción de entidades

| **ID** | **Nombre** | **Descripción** |
| --- | --- | --- |
| **E1** | HOME | Representa una vivienda u oficina dentro de la urbanización o proyecto de construcción. |
| **E2** | CONTACT\_HOME | Representa la asociación entre los contactos y las viviendas u oficinas. Permite conocer cuáles son los contactos asociados a una vivienda u oficina. |
| **E3** | CONTACT | Representa un contacto que puede tener una vivienda u oficina. |
| **E4** | EVENT | Representa un evento generado dentro del sistema por alguno de los sensores de las viviendas u oficinas. |
| **E5** | ITEM | Representa un sensor perteneciente a alguna de las casas que permite controlar los eventos que ocurren dentro de esta. |
| **E6** | RULE | Representa una regla que es configurada sobre alguno de los ítems. |
| **E7** | USER | Representa un usuario del sistema. |
| **E8** | USER\_ROLE | Representa la asociación entre los roles y los usuarios. Permite identificar los roles asignados al usuario. |
| **E9** | ROLE | Representa un rol que puede ser configurado a los usuarios del sistema. |
| **E10** | ROLE\_FUNCTION | Representa la asociación entre los roles y las funciones. Permite conocer que funciones puede realizar un rol dentro del sistema. |
| **E11** | FUNCTION | Representa una función que puede ser realizada por un usuario dentro del sistema. |
| **E12** | AUDIT | Representa una función que ha sido realizada dentro del sistema. Permite realizar auditoria sobre las acciones dentro del sistema como la fecha y el usuario que la realizó. |

* + 1. **Modelos de Flujo de Información**

En el modelo de flujo de información veremos la interacción de los datos de acuerdo a los procesos de negocio. Los modelos descritos a continuación se realizaron para los procesos más importantes, de tal forma que se pueda representar el flujo de las entidades del sistema.

**Evento**

Se describe el proceso que se realiza al encenderse la alarma de incendio.

Tabla 36. Modelo de flujo de información: Evento

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | | | | | **Familia**  Module ( )  C&C (X )  Allocation ( )  **Estilo arquitectural**  Coordinador  Cache |
| **Convención**    Proceso de datos    Entidad de datos  Relación que indica la dirección del flujo de datos |
| **Título:**  Modelo de flujo de información: Evento | | **ID:**  PV03 | **Nivel Profundidad:**  Nivel 1 | | **Nomenclatura:**  Diagrama de flujo |
| **Arquitecto:**  Erik Arcos | **Grupo:**  INGENIUM | | | **Fecha:**  09/07/2011 | **Versión:**  1.0 |

El sensor del detector de humo genera un evento que es recibido por el sistema central el cual se encarga de consultar la información del sensor o ítem, persistir la información del evento y continuar con el procesamiento del evento de acuerdo al tipo de evento.

En la parte de procesamiento se debe consultar las reglas asociadas al sensor y con base en estas reglas se decide si se debe realizar la notificación o terminar el proceso. Si se requiere la notificación se debe consultar la información de los contactos asociados a la casa que donde se presenta el incendio.

**Regla de item**

Se describe el proceso que se realiza cuando un propietario de una vivienda u oficina parametriza la regla para un sensor específico.

Tabla 37. Modelo de flujo de información: Regla item

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | | | | | **Familia**  Module ( )  C&C ( X )  Allocation ( )  **Estilo arquitectural**  Cache  PACE |
| **Convención**    Proceso de datos    Entidad de datos  Relación que indica la dirección del flujo de datos |
| **Título:**  Modelo de flujo de información: Regla item | | **ID:**  PV03 | **Nivel Profundidad:**  Nivel 1 | | **Nomenclatura:**  Diagrama de flujo |
| **Arquitecto:**  Sandra Gómez | **Grupo:**  INGENIUM | | | **Fecha:**  09/07/2011 | **Versión:**  1.0 |

El propietario ingresa a la aplicación por medio un usuario y contraseña que son validados con la información almacenada en base de datos, y se obtiene la información de los roles asociados. Luego ingresa a la opción de parametrizar regla de sensor, y se valida contra la base de datos que la función este permitida para alguno de los roles asociados al usuario, si es así se registra o actualiza la información de la regla asociada al sensor, y se registra una auditoria sobre la acción realizada en el sistema. Finalmente se actualiza el cache que contiene las reglas de los ítems.

* + 1. **Modelos de Ciclo de Vida de Información**

En este ciclo de vida describiremos los posibles estados por los que puede pasar un evento.

Tabla 38. Modelo de ciclo de vida de información

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | | | | | **Familia**  Module ( )  C&C ( X )  Allocation ( )  **Estilo arquitectural**  Cache |
| **Convención**    Estado inicial    Estado final  Transición    Decisión    Estado final |
| **Título:**  Modelo de ciclo de vida de información | | **ID:**  PV03 | **Nivel Profundidad:**  Nivel 4 | | **Nomenclatura:**  Diagrama de estados |
| **Arquitecto:**  Sandra Gómez | **Grupo:**  INGENIUM | | | **Fecha:**  0/07/2011 | **Versión:**  1.0 |

Al encenderse la alarma de incendio se genera un evento nuevo, este evento es enviado por el concentrador que existe en cada vivienda u oficina, luego el evento es recibido por el sistema central y se valida la autenticidad del evento para después pasar a analizar el evento de acuerdo a las reglas configuradas para el sensor. Del análisis depende si el evento es procesado para su notificación o se descarta.

* 1. **Punto de Vista de Concurrencia**
     1. **Descripción**

En este punto vista describe la concurrencia del sistema, mostrando los elementos funcionales de los sistemas que se están comunicando.

* + 1. **Modelos**

Para este documento se analiza los sistemas ASHHomeModule y ASHLoadBalancer.

Tabla 39. Modelo de concurrencia del sistema

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | | | | | **Familia**  Module ( X )  C&C ( )  Allocation ( )  **Estilo arquitectural**  Redundancia active  PACE |
| **Convención**  Relación |
| **Título:**  Modelo de concurrencia del sistema | | **ID:**  PV01 | **Nivel Profundidad:** | | **Nomenclatura:** |
| **Arquitecto:**  Sandra Gómez | **Grupo:**  INGENIUM | | | **Fecha:**  29/06/2011 | **Versión:**  1.0 |

* 1. **Punto de Vista de Desarrollo**
     1. **Descripción**

En este punto de vista se aborda la estructura de código y las dependencias del proyecto Alpes Smart Home. De esta manera podemos observar que paquetes y/o artefactos hicieron parte del proceso de desarrollo del sistema para llevarlo a su estado final, el cual se presenta a continuación en forma de un modelo de estructura de módulos.

* + 1. **Modelos**

Tabla 40. Modelo de desarrollo del sistema

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | | | | | **Familia**  Module ( X )  C&C ( )  Allocation ( )  **Estilo arquitectural**  PACE  Cache  Capas |
| **Convención** |
| **Título:**  Modelo de desarrollo del sistema | | **ID:**  PV01 | **Nivel Profundidad:** | | **Nomenclatura:** |
| **Arquitecto:**  Mauricio Erazo | **Grupo:**  INGENIUM | | | **Fecha:**  29/06/2011 | **Versión:**  1.0 |

Los módulos descritos en este modelo representan los paquetes de la estructura de código que soportan las funcionalidades del proyecto. Está dividido en dos partes. La primera representa toda la estructura del sistema encargado de enviar y recibir los eventos que se generan en los domicilios, y está conformado por distintos paquetes, los cuales son ASHHomeModule, ASHCentralSystem, ASHLoadBalancer, ASHCommons, ASHTestSystem, ASHExternalSystem. La segunda parte representa el sistema Web encargado de presentar al usuario la información del sistema.

El paquete ASHHomeModule contiene una serie de paquetes, los cuales en conjunto se encargan de simular el funcionamiento del dispositivo de hardware que se encuentra en los domicilios y que es encargado de enviar los eventos del sistema hasta el sistema central para su posterior procesamiento. En este paquete se pueden destacar los paquetes **ash,** el cual contiene los certificados de firmas digitales usados para firmar las tramas que se envían al sistema con el fin de asegurar la integridad de los datos. También se destaca el paquete **security,**  el cual se encarga de cifrar la información enviada con el fin de asegurar la confidencialidad de los datos enviados a través del canal. El resto de paquetes son los encargados de realizar la conexión a través de sockets hacia el sistema central y de enviar las tramas que contienen los datos de los eventos. En este paquete también se encuentra el paquete **º** que se encarga de realizar la video llamada a través de Skype desde un computador ubicado en la casa, el cual dispone de una cámara Web y el cual se encuentra previamente logeado en una cuenta de Skype, así como de enviar un correo de información en caso de que la llamada no sea contestada.

El paquete ASHLoadBalancer es aquel que tiene la conexión directa con el sistema en el domicilio (**ASHHomeSystem**) y tiene la funcionalidad necesaria para verificar las firmas digitales y realizar el desciframiento del mensaje. También tiene la funcionalidad necesaria para repartir las peticiones entrantes a las diferentes instancias del sistema central, tal como se puede observar en el diagrama de despliegue.

El paquete ASHCentralSystem es aquel que alberga la funcionalidad core del sistema. Este se encarga de recibir los eventos desde el balanceador de cargas a través del paquete **communication**, así mismo alberga las funcionalidades necesarias para generar y actualizar el cache con la información necesaria de los usuarios y sus configuraciones (paquete cache). El paquete manager es el encargado de determinar qué tipo de evento se ha generado y de esta manera determinar la acción que debe realizarse. Se tiene un paquete con los objetos de negocio (**paquete bo**), el cual alberga los objetos necesarios para mapear la información de los clientes que luego será persistida en la base de datos. También se destaca el paquete **notificator**, el cual es el encargado de notificar a los clientes sobre los eventos que están ocurriendo en el domicilio. Los otros paquetes se encargan de funciones básicas del sistema, como por ejemplo el arranque del mismo y la comunicación entre los otros paquetes. Un paquete que vale la pena resaltar es el de **test**, el cual, aunque no es usado directamente por el sistema, es el que permitió realizar las pruebas de carga del sistema para verificar que se cumplen los atributos de calidad requeridos.

El paquete ExternalSystem se desarrolló con el objeto de simular la conexión con sistemas externos de emergencia, los cuales pueden ser policía, emergencias, etc., y realiza una conexión a través de sockets con un sistema que previamente debe ser configurado.

En el paquete ASHCommons contiene un paquete de utilidades (util), el cual contra de una clase con las contantes comunes, y una clase de utilidades genéricas. Como se puede ver en el diagrama, este paquete es usado por todos los paquetes del core del sistema, pues provee los datos y funciones necesarias que son transversales a todo el sistema.

Finalmente tenemos el paquete ASHTestSystem, el cual se integra y hace uso de todos los paquetes previamente nombrados, pues este se encarga de realizar todos los test correspondientes para verificar que se están cumpliendo los diferentes atributos y escenarios de calidad especificados, por ejemplo, un test de carga sobre el balanceador, test de denegación de servicio, test de envío y recepción de mensajes correctamente conformados, test de envío y recepción de mensajes erróneos, etc.

Tenemos otro paquete representativo, y es el paquete correspondiente a todo el modulo Web ASHWebModule, el cual contiene todas los paquetes necesarios para proveer las funcionalidades necesarias para el correcto funcionamiento del módulo web, el cual se encarga de mostrar la información del sistema al usuario, así como de modificar sus preferencias. Consta de varios paquetes. Entre ellos el paquete **entities**, el cual contiene las entidades del sistema necesarias para albergar y desplegar la información del mismo, el paquete **facade** el cual encapsula las funciones necesarias para administrar estas entidades, que se accederán a través de las fachadas dispuestas en el paquete **bean.** También tiene dos paquetes de utilidades encargados de realizar la paginación del sistema, el login al sistema, la des encriptación por medio del algoritmo MD5, y otras funciones comunes y genéricas del sistema. También tenemos el paquete **reporte** el cual el responsable de generar el reporte que se mostrara al usuario final consultando la información que se ha persistido en la base de datos conectándose a la misma a través de las funciones contenidas en el paquete **connection.**

1. **Relaciones entre los Puntos de Vista**

Describiremos la relación que hay entre los diferentes puntos de vista y la consistencia que hay entre ellas.

* 1. **Punto de vista funcional – Punto de vista de despliegue**

Es importante conocer como la parte funcional sobre que componente se permite su funcionamiento, para esto a continuación se muestra una relación entre los componentes de la vista funcional y los componentes de la vista de despliegue.

Tabla 41. P.V. Despliegue y P.V. Funcional

|  |  |  |
| --- | --- | --- |
| **Componentes P.V. Despliegue** | **Componentes P.V. Funcional** | **Descripción** |
| Main Server  Data Server | ASHCentralSystem | Tanto el servidos principal y servidor de datos soportan el funcionamiento del sistema central que permite el procesamiento de los datos. |
| Load Balance Server | ASHLoadBalancer | El servidor para el balanceo de carga es quien soporta la aplicación que se encarga de repartir las peticiones que llegan. |
| Device Home | ASHHomeSystem | El dispositivo de la vivienda soporta una pequeña aplicación que concentra todas las peticiones enviadas por los diferentes sensores de la vivienda y enviárselas al servidor principal. |
| Web Server  Data Server | ASHWebModule | El servidor Web y el servidor de datos soportan la aplicación Web que permite realizar las consultas necesarias para obtener los reportes de estado, además puede editar y consultar la información para la administración de usuarios y configuración sobre los diferentes dispositivos de la vivienda. |

* 1. **Punto de vista funcional – Punto de vista de información**

|  |  |  |
| --- | --- | --- |
| **Componentes P.V. Funcional** | **Componentes P.V. Información** | **Descripción** |
| ASHCentralSystem | EVENT  HOME  CONTACT\_HOME  CONTACT  ITEM  RULE | Para la operación del sistema central se requieren todos los datos asociados a los eventos generados por las diferentes casas y los datos asociados referentes a las acciones que debe tomar al recibir un evento. |
| ASHWebModule | HOME  EVENT  ITEM  RULE  USER  FUNCTION  ROLE\_FUNCTION  ROLE  USER\_ROLE  AUDIT | En la aplicación Web se requiere información que permita generar eventos y contralar los diferentes dispositivos de las casas, además de controlar la parte de seguridad de usuarios. |

* 1. **Punto de vista funcional – Punto de vista de desarrollo**

|  |  |  |
| --- | --- | --- |
| **Componentes P.V. Funcional** | **Componentes P.V. Desarrollo** | **Descripción** |
| ASHCentralSystem | Contiene los paquetes:  test  control  notificator  manager  bo  cache  communication  manager  valueobjects | El sistema central contiene todo el desarrollo que permite procesar los diferentes eventos que llegan desde las viviendas oficinas, permitiendo notificar, de acuerdo a las reglas configuradas por cada dispositivo, a las diferentes entidades externas como policía o bomberos. |
| ASHLoadBalancer | Ash  Load  security | El balanceador de carga contiene el desarrollo que permite repartir las peticiones que llegan desde las viviendas u oficinas, y también verifica las firmas de la información recibida. |
| ASHHomeSystem | ash  communication  control  notificator  security  skype.resources  vo | El sistema de la casa funciona como un concentrador de la información enviada por todos los sensores de la vivienda. También permite realizar video llamadas cuando se presenta una alerta por el sensor de movimiento. |
| ASHWebModule | Contiene los paquetes:  bean  entities  facade  transportdate  facade.util  report  util  connection | Contiene toda la parte de desarrollo que permite al usuario generar reportes y parametrizar reglas de dispositivos. |

* 1. **Punto de vista despliegue– Punto de vista de información**

Todas las entidades del sistema están soportadas por el servidor de datos que contiene la base de datos Oracle.

1. **Evaluación de la Arquitectura**
   1. **Atam**

A través de todo el proceso de desarrollo del sistema se generaron diferentes requerimientos que debían ser cumplidos, pero así mismo se generaron diferentes inquietudes que debían ser resueltas y obstáculos que debían ser superados para lograr cumplir con las expectativas que esperaban los StakeHolders del sistema.

Durante el primer experimento del sistema se realizó el enfoque primario de la arquitectura en base al desempeño, pues este era el atributo de calidad de mayor relevancia para los StakeHolders. Una vez identificado que este atributo seria el principal del sistema se procedió a tomar las decisiones de arquitectura necesarias para satisfacerlo. Algunas de estas decisiones incluyeron el manejo de las tramas bit a bit, el uso del patrón de arquitectura Lazy Adquisition, establecer la comunicaciones entre los sistemas por medio de Sockets, etc., todo con el fin de cumplir con los escenarios operaciones requeridos tratando de dejar el margen más amplio posible para que al momento de incluir los siguientes atributos de calidad no se llegara a la eventualidad de no cumplir con el desempeño requerido.

Cuando se definió el segundo experimento, el cual estaba enfocado en la seguridad y disponibilidad de la aplicación, también se tomaron decisiones de arquitectura, entre las cuales se cuentan el uso de firmas digitales con el propósito de asegurar la integridad de los datos, así como algoritmos de encriptación para asegurar la confidencialidad de los mismos. Se incluyó un balanceador de carga con el fin de favorecer la disponibilidad del sistema que se encargaría de distribuir las peticiones a diferentes instancias del sistema central, evitando así un fallo general del sistema. Se identificaron posibles puntos sensibles de falla y cuellos de botella, por ejemplo, que la seguridad del sistema se manejara en un único componente, en base a esto se pudo determinar que una de las posibles mejoras al sistema seria incluir el componente de seguridad en cada una de las instancias del sistema central. Obviamente todas estas decisiones tuvieron un impacto en el de desempeño del sistema, pero debido a que previamente se había considerado esto, se pudo seguir cumpliendo los escenarios de calidad de desempeño.

El tercer experimento nos llevó por el camino de la interoperabilidad del sistema, es decir, que nuestro sistema fuera capaz de comunicarse con otros sistemas y usar otras aplicaciones. Debido a las decisiones que se habían tomado en las etapas anteriores del desarrollo de la arquitectura del sistema, la inclusión de la interoperabilidad no significo grandes cambios en la arquitectura previamente definida. Se debió incluir las funcionalidades necesarias para hacer la comunicación con WebServices externos, y se realizó el desarrollo necesario para que el sistema pudiera realizar una video llamada como respuesta a un evento especifico. Uno de los cambios radicales que se presentó en el sistema, mas a nivel de despliegue, que a nivel arquitectural o de desarrollo fue el hecho de que se debía contar con un computador en el domicilio al cual estaría conectada la o las videocámaras y desde el cual se realizaría la video llamada a un usuario previamente definido en la administración del sistema. Esto, obviamente implica un leve aumento en los costos que tendrían que pagar los usuarios finales, pero si tomamos en cuenta que este tipo de sistemas está orientado para personas cuyo poder adquisitivo es alto, podríamos decir que este aumento en el costo no impacta en gran cuantía los motivadores de negocio actuales.

Al final se obtuvo un sistema integrado, en el cual se cumplió con los atributos y escenarios de calidad inicialmente definidos, satisfaciendo así las necesidades de los StakeHolders involucrados.

* 1. **Experimentación**

A continuación se presentan los resultados obtenidos de lasexperimentaciones con lo cual se tomaron decisiones para la arquitectura de solución.

* + 1. **Experimento 1 - Desempeño**

* + - 1. **Experimento A – Latencia**

El objetivo de este experimento es determinar si las decisiones de diseño satisfacen los escenarios de calidad relacionados con la latencia de la aplicación.

* + - * 1. **A1 – Registro de Alarma**

Tabla 42. Descripción del Experimento 1 - A1

|  |  |
| --- | --- |
| **Descripción del Experimento** | |
| ***Título*:** Registro de Alarma | ***ID*:**1 -A1 |
| ***Descripción*:** Este experimento busca verificar la latencia del sistema | ***Responsable*:**Ingenium |
| ***Propósito:***  ( ) Reparación, actualizar, clarificar  ( X ) Obtener Información técnica  ( ) Obtener información de negocio  ( ) Otros: | |
| ***Propósito:***  Verificar las decisiones de diseño asociadas a la latencia del sistema. | |
| ***Descripción del experimento:***  El sistema debe estar en capacidad de analizar la información enviada por los sensores de humo, ventanas y puertas de las oficinas y viviendas monitoreadas.  Se debe tener en cuenta lo siguiente:   * Una vez se detecta humo en una posición específica de la vivienda u oficina, se procede a notificar a los servicios de control de emergencias. * Si se trata de un sensor de ventana o puerta abierta sin autorización, se debe notificar a la policía sobre el evento, enviando como información adicional, la localización de la ventana dentro del inmueble y los sectores de la vivienda en los que se ha detectado movimiento. | |
| ***Artefactos Creados:***  ASHCentralSystem ->Es el proyecto principal donde se encuentra toda la lógica de negocio para procesar los eventos recibidos por cada casa u oficina.  ASHCommunication -> Librería que administra la comunicación entre el sistema central y el concentrador ubicado en cada casa u oficina.  ASHHomeModule ->Es el proyecto con la lógica para la simulación de los eventos generados por la casa u oficina.  ASHExternalSistem -> Sistema que emula la policía y los sistemas de emergencia.  Para la ejecución de esta parte del experimento se empleó como mensaje una trama definida como se muestra en la siguiente gráfica:  **Trama**  **Evento**  La trama transmite la información relacionada de los eventos detectados en cada vivienda. A su vez el evento contiene la información del tipo, código y estado del sensor de la vivienda. | |
| ***Criterio de terminación:***  El tiempo para la notificación de la información, una vez recibida la alarma, no debe ser superior a 1 segundo. | |
| ***Recursos Requeridos:***  Recurso humano -> Se necesita el trabajo de 4 personas con conocimientos en java, sockets, estilos y tácticas de arquitectura.  Recursos técnicos ->NetBeans IDE 7.0, Microsoft Visio, JUnit, JUnitPerf | |
| ***Duración estimada:***  32 horas | |

Tabla 43. Resultados del Experimento 1 - A1

|  |
| --- |
| **Resultados del Experimento** |
| ***Resumen de los resultados:***  Se implemento la arquitectura descrita anteriormente y se ejecuto el test de envío de información desde una casa.  Se desarrollo un test que permite emular el envío del estado de los 50 sensores de una casa (Humo, Puertas, RFID), el estado de todos estos sensores es enviado periódicamente cada segundo.  El test se ejecuto 10 veces dando como resultado un promedio de 436ms desde el momento en que el sistema central recibe el mensaje y es interpretado para notificar a las autoridades correspondientes. |
| ***Duración Real:***  50 horas |
| ***Recursos Reales:***  Recurso humano -> Se necesita el trabajo de 3 personas con conocimientos en java, estilos y tácticas de arquitectura, bases de datos y diseño Web.  Recursos técnicos ->NetBeans IDE 7.0, Microsoft Visio |
| ***Recomendaciones:***  Se había implementado inicialmente una notificación por medio de e-mail al usuario, el envío del correo ya consumía alrededor de 10s, por lo tanto esta notificación fue eliminada y queda pendiente de análisis para ser reemplazada.  La comunicación con las autoridades se realiza por medio de sockets directamente a sus centrales correspondientes. |

* + - * 1. **A2 – Consulta de Información Consolidada**

Tabla 44. Descripción del Experimento 1 - A2\_1

|  |  |
| --- | --- |
| **Descripción del Experimento** | |
| ***Título*:** Latencia | ***ID*:** 1 – A2\_1 |
| ***Descripción*:** Este experimento busca verificar el desempeño del sistema en el generación de reporte consolidado de eventos generados. | ***Responsable*:** Ingenium |
| ***Propósito:***  ( ) Reparación, actualizar, clarificar  ( X ) Obtener Información técnica  ( ) Obtener información de negocio  ( ) Otros: | |
| ***Propósito:***  Verificar las decisiones de diseño asociada a la latencia del sistema. | |
| ***Descripción del experimento:***  Verificar el tiempo necesario para procesar la solicitud de un reporte consolidado de los 400 eventos registrados (ordenadas por tipo de evento) de forma cronológica (hora, día, mes, año). | |
| ***Artefactos Creados:***  ASHWebModule ->Sistema Web que permite el procesamiento para generar el reporte consolidado de eventos presentados ante el sistema central. Posee una conexión con la base de datos de donde obtiene la información de los eventos. | |
| ***Criterio de terminación:***  Este reporte no debe tomar más de 2 segundos en ser presentado al usuario. | |
| ***Recursos Requeridos:***  Recurso humano -> Se necesita el trabajo de 2 personas con conocimientos en java, estilos y tácticas de arquitectura, bases de datos y diseño Web.  Recursos técnicos -> Base de datos Oracle XE, NetBeans IDE 7.0, Microsoft Visio, Navegador, editor SQL Developer | |
| ***Duración estimada:***  16 horas | |

Tabla 45. Resultados del Experimento 1 - A2\_1

|  |
| --- |
| **Resultados del Experimento** |
| ***Resumen de los resultados:***  Se realizaron pruebas por medio de la Suite de pruebas de Microsoft Visual Studio, generando el reporte con 400 eventos registrados en base de datos, en un tiempo de 0,442 segundos. |
| ***Duración Real:***  19 horas |
| ***Recursos Reales:***  Recurso humano -> Se necesita el trabajo de 3 personas con conocimientos en java, estilos y tácticas de arquitectura, bases de datos y diseño Web.  Recursos técnicos -> Base de datos Oracle XE, NetBeans IDE 7.0, Microsoft Visio, Navegador, editor SQL Developer |
| ***Recomendaciones:***  Verificar el experimento A2-2 |

Tabla 46. Descripción del Experimento 1 - A2\_2

|  |  |
| --- | --- |
| **Descripción del Experimento** | |
| ***Título*:** Latencia | ***ID*:**1 - A2\_2 |
| ***Descripción*:** Este experimento busca verificar el desempeño del sistema en el generación de reporte consolidado de eventos generados. | ***Responsable*:** Ingenium |
| ***Propósito:***  ( ) Reparación, actualizar, clarificar  ( X ) Obtener Información técnica  ( ) Obtener información de negocio  ( ) Otros: | |
| ***Propósito:***  Verificar las decisiones de diseño asociada a la latencia del sistema. | |
| ***Descripción del experimento:***  Verificar el tiempo necesario para procesar la solicitud de un reporte consolidado de los 4800 eventos registrados (ordenadas por tipo de evento) de forma cronológica (hora, día, mes, año). | |
| ***Artefactos Creados:***  ASHWebModule -> Sistema Web que permite el procesamiento para generar el reporte consolidado de eventos presentados ante el sistema central. Posee una conexión con la base de datos de donde obtiene la información de los eventos. | |
| ***Criterio de terminación:***  Este reporte no debe tomar más de 2 segundos en ser presentado al usuario. | |
| ***Recursos Requeridos:***  Recurso humano -> Se necesita el trabajo de 2 personas con conocimientos en java, estilos y tácticas de arquitectura, bases de datos y diseño Web.  Recursos técnicos -> Base de datos Oracle XE, NetBeans IDE 7.0, Microsoft Visio, Navegador, editor SQL Developer | |
| ***Duración estimada:***  2 horas | |

Tabla 47. Resultados del Experimento 1 - A2\_2

|  |
| --- |
| **Resultados del Experimento** |
| ***Resumen de los resultados:***  Se realizaron pruebas por medio de la Suite de pruebas de Microsoft Visual Studio, generando el reporte con 4800 eventos registrados en base de datos, en un tiempo de 4,165 segundos (Ver Recomendaciones).  Es importante anotar que el tiempo se aumentó con relación al experimento A2-1 debido al tiempo adicional tomado por el navegador en realizar el renderizado de todos los datos La información sale de la base de datos en 0,6 segundos. |
| ***Duración Real:***  3 horas |
| ***Recursos Reales:***  Recurso humano -> Se necesita el trabajo de 3 personas con conocimientos en java, estilos y tácticas de arquitectura, bases de datos y diseño Web.  Recursos técnicos -> Base de datos Oracle XE, NetBeans IDE 7.0, Microsoft Visio, Navegador, editor SQL Developer |
| ***Recomendaciones:***  Para mejorar el rendimiento del tiempo en la generación del reporte es recomendable realizar una paginación en presentación y realizar paginación por base de datos, es decir, utilizar el patrón Acceso a recursos temprana (LazyAcquisition). Esta recomendación puede tomar valor en el crecimiento de registros en base de datos de los eventos generados. |

* + - 1. **Experimento B – Escalabilidad**

El objetivo de este experimento es determinar la escalabilidad del sistema. Para ello usted deberá utilizar una herramienta para análisis de desempeño y escalabilidad como JUnitPerf, para simular la ocurrencia de múltiples eventos concurrentemente.

* + - * 1. **B1 – Escalabilidad del Sistema**

Tabla 48.Descripción del Experimento 1 - B1

|  |  |
| --- | --- |
| **Descripción del Experimento** | |
| ***Título*:** Escalabilidad | ***ID*:**1 - B1 |
| ***Descripción*:** Este experimento busca verificar la escalabilidad del sistema | ***Responsable*:** Ingenium |
| ***Propósito:***  ( ) Reparación, actualizar, clarificar  ( X ) Obtener Información técnica  ( ) Obtener información de negocio  ( ) Otros: | |
| ***Propósito:***  Verificar las decisiones de diseño asociadas a la escalabilidad del sistema. | |
| ***Descripción del experimento:***  En este experimento debe simular la recepción de información proveniente de los tags RFID que se encuentran en cada uno de los inmuebles. Suponga que por cada inmueble u oficina se tienen en promedio 50 tags RFID enviando información cada segundo a cuatro antenas ubicadas en diferentes posiciones de la casa u oficina. Estas antenas a su vez consolidan la información y la envían a la central. | |
| ***Artefactos Creados:***  ASHCentralSystem ->Es el proyecto principal donde se encuentra toda la lógica de negocio para procesar los eventos recibidos por cada casa u oficina.  ASHCommunication -> Librería que administra la comunicación entre el sistema central y el concentrador ubicado en cada casa u oficina.  ASHHomeModule ->Es el proyecto con la lógica para la simulación de los eventos generados por la casa u oficina.  ASHExternalSistem -> Sistema que emula la policía y los sistemas de emergencia. | |
| ***Criterio de terminación:***  Cuando se detecta que un tag RFID ha violado una de las reglas definidas para su comportamiento, se debe generar una alarma indicando a todas las porterías y a los responsables de la vivienda u oficina lo ocurrido. Esto debe suceder en un tiempo inferior a un segundo. | |
| ***Recursos Requeridos:***  Recurso humano -> Se necesita el trabajo de 3 personas con conocimientos en java, sockets, estilos y tácticas de arquitectura.  Recursos técnicos ->NetBeans IDE 7.0, Microsoft Visio, JUnit, JUnitPerf | |
| ***Duración estimada:***  8 horas | |

Tabla 49. Resultados del Experimento 1 - B1

|  |
| --- |
| **Resultados del Experimento** |
| ***Resumen de los resultados:***  A continuación se presenta la gráfica de tiempo contra cantidad de usuarios concurrentes. Se puede notar que a medida que se incrementa el número de usuarios concurrentes, el tiempo de respuesta aumenta cada vez más rápido.    En la siguiente gráfica se puede notar como el número de fallas (una falla equivale a que un evento se procesó con un tiempo superior a un segundo) no aumenta hasta llegar a un punto donde se dispara repentinamente (entre 2400 a 2600 usuarios concurrentes). Este punto es importante, porque es el que determina la capacidad que tiene el sistema de responder a un número de usuarios concurrentes manteniendo las reglas del negocio (un segundo). |
| ***Duración Real:***  12 horas |
| ***Recursos Reales:***  Recurso humano -> Se necesita el trabajo de 2 personas con conocimientos en java, sockets, estilos y tácticas de arquitectura.  Recursos técnicos ->NetBeans IDE 7.0, Microsoft Visio, JUnit, JUnitPerf |
| ***Recomendaciones:***  Cuando se requiera aumentar el número de usuarios concurrentes más allá de lo que puede responder el sistema, se puede hacer uso de tácticas de escalabilidad / desempeño como por ejemplo el balanceo de cargas. |

* + 1. **Experimento 2 – Seguridad y Disponibilidad**
       1. **Experimento A – Seguridad**

El objetivo de este experimento es validar las decisiones de diseño tomadas para favorecer la seguridad tanto en la generación del reporte como en el envío de tramas desde los dispositivos del hogar hasta el sistema central.

* + - * 1. **A1 – Autenticación y Autorización**

Tabla 50. Descripción del Experimento 2 - A1

|  |  |
| --- | --- |
| **Descripción del Experimento** | |
| ***Título*:** Autenticación y autorización | ***ID*:**2 - A1 |
| ***Descripción*:** Este experimento busca aplicar las tácticas de autenticación y autorización para evitar modificación y consulta de información no autorizada. | ***Responsable*:** Ingenium |
| ***Propósito:***  ( ) Reparación, actualizar, clarificar  ( X ) Obtener Información técnica  ( ) Obtener información de negocio  ( ) Otros: | |
| ***Propósito:***  Validar los desarrollos del sistema de autenticación, autorización para la modificación de datos relacionado a los usuarios. | |
| ***Descripción del experimento:***  El sistema debe ser capaz de:   * Registrar nuevos usuarios con su nombre de usuario, password y rol asociado. * Validar el ingreso de un usuario registrado. * Determinar las acciones y permisos para cada usuario. | |
| ***Artefactos Creados:***  ASHWebSystemAuthentication -> Es el proyecto donde se valida la autenticación para el ingreso del usuario y el cifrado Hash.  ASHWebSystemAuthorization -> Autorización para la ejecución de funciones al rol asociado.  ASHWebSystem -> Se encarga de creación, edición de usuarios del sistema. | |
| ***Criterio de terminación:***  Un usuario no registrado en el sistema no debe poder acceder las funciones disponibles (generación de reporte).  El usuario no autorizado no puede acceder a las funciones a las cuales no tenga permiso. | |
| ***Recursos Requeridos:***  Recurso humano -> Se necesita el trabajo de 2 personas con conocimientos en java, JEE, JSF, autorización y autenticación.  Recursos técnicos ->NetBeans IDE 7.0, Microsoft Visio, Selenium IDE 1.0.11 | |
| ***Duración estimada:***  13 horas | |

Tabla 51. Resultados del Experimento 2 - A1

|  |
| --- |
| **Resultados del Experimento** |
| ***Resumen de los resultados:***  Se implementaron pruebas en la herramienta Selenium IDE para probar el funcionamiento del sistema de autenticación y autorización.  En el primer caso de prueba se intenta acceder con datos de usuario incorrectos, arrojando un mensaje de error en los datos ingresados.    En el segundo caso de prueba se accede con un usuario que no tiene permiso de visualización de la información de reportes de alertas, mostrando al usuario que no tienen autorización a dicha información.    En la tercera prueba, Se accede con usuario que tiene permiso de visualización del reporte y se puede acceder a dicha información.    En este experimento se obtuvo un tiempo de respuesta inferior a 2 segundos. |
| ***Duración Real:***  14 horas |
| ***Recursos Reales:***  Recurso humano -> Se necesita el trabajo de 2 personas con conocimientos en java, JEE, JSF, autorización y autenticación.  Recursos técnicos ->NetBeans IDE 7.0, Microsoft Visio, Selenium IDE 1.0.11 |
| ***Recomendaciones:***  Se puede hacer uso de las herramientas dispuestas para seguridad y autorización de las tecnologías Java, haciendo un manejo de sesiones más sencillo y dejando el manejo de la autenticación y autorización al contenedor de aplicaciones. |

* + - * 1. **A2 – Denegación de Servicio / Suplantación**

Tabla 52. Descripción del Experimento A2

|  |  |
| --- | --- |
| **Descripción del Experimento** | |
| ***Título*:**Denegación de servicio / suplantación | ***ID*:**A2 |
| ***Descripción*:**  Este experimento busca verificar que el sistema se protege contra un ataque de denegación de servicios (DoS) y suplantación de información. | ***Responsable*:** Ingenium |
| ***Propósito:***  ( ) Reparación, actualizar, clarificar  ( X ) Obtener Información técnica  ( ) Obtener información de negocio  ( ) Otros: | |
| ***Propósito:***  Verificar y validar los artefactos desarrollados y la arquitectura planteada en términos de seguridad y disponibilidad. | |
| ***Descripción del experimento:***  Verificar el desarrollo para el sistema de seguridad es capaz de detectar y soportar un ataque de denegación de servicios. Detectar si la información proveniente de una casa es legítima.  Se simulan dos posibles ataques al sistema, el primero un atacante que envía peticiones al sistema con mayor frecuencia que los sistemas de las casas normales, el sistema lo detecta y evita el procesamiento de sus peticiones. El segundo ataque simula un mensaje al sistema central que no ha sido firmado correctamente, el sistema lo detecta y evita procesar este mensaje. En ambos casos se notifica y se cierra la conexión con la casa para evitar la sobrecarga del sistema. | |
| ***Artefactos Creados:***  ASHLoadBalancerBlackList -> El sistema permite una detección de un ataque de denegación de servicios de acuerdo a la frecuencia de llegada de peticiones provenientes de una vivienda, registrando la fuente del ataque y evitando el procesamiento de las peticiones hasta que manualmente sea removido del BlackList.  ASHHomeSystemAuthentication -> Permite verificar que la información proviene de una fuente legitima. Se encarga de establecer una firma desde el sistema Home System para que sea verificada en el sistema de seguridad desde el módulo Home SystemAutentication. En caso de no verificarse satisfactoriamente, se notifica y se cierra la conexión con esa casa para evitar ataque DOS. | |
| ***Criterio de terminación:***  El sistema detecta los ataques de denegación de servicio y los controla, evitando procesar más peticiones de estas fuentes y avisando acerca del ataque para que sea atendido. | |
| ***Recursos Requeridos:***  Recurso humano -> Se necesita el trabajo de 2 personas con conocimientos en java, estilos y tácticas de arquitectura, bases de datos, sockets, Threads, etc.  Recursos técnicos -> Base de datos Oracle XE, NetBeans IDE 7.0, Microsoft Visio, JUnit. | |
| ***Duración estimada:***  16 horas | |

Tabla 53. Resultados del Experimento 2 - A2

|  |
| --- |
| **Resultados del Experimento** |
| ***Resumen de los resultados:***  En la implementación de tácticas como firmas digitales se obtuvo el siguiente resultado.   * Si la trama enviada desde la casa/oficina no se verifica adecuadamente, dicha casa/oficina será bloqueada y se realizará la notificación respectiva. * Si existe alguna casa/oficina que envíe tramas con una frecuencia superior a la normal, dicha casa/oficina fue bloqueada y notificado dicho evento. |
| ***Duración Real:***  22 horas |
| ***Recursos Reales:***  Recurso humano -> Se necesita el trabajo de 2 personas con conocimientos en java, estilos y tácticas de arquitectura, bases de datos, sockets, Threads, etc.  Recursos técnicos -> Base de datos Oracle XE, NetBeans IDE 7.0, Microsoft Visio, JUnit. |
| ***Recomendaciones:***  Se recomienda ubicar los mecanismos de seguridad (verificación de firmas) en cada uno de los sistemas centrales existentes, debido a que como se encuentra en el momento (en el servidor del balanceador de cargas) se convierte en un cuello de botella al momento de verificar las firmas digitales provenientes de cada casa/oficina. |

* + - 1. **Experimento B – Disponibilidad**

El objetivo de este experimento es validar la decisión de diseño que involucra la implementación de un balanceador de cargas en el sistema. Además se requiere demostrar que los tiempos aún se conservan después de satisfacer atributos de calidad como seguridad (incluido en este experimento)

* + - * 1. **B – Disponibilidad**

Tabla 54. Descripción del Experimento 2 - B

|  |  |
| --- | --- |
| **Descripción del Experimento** | |
| ***Título*:** Disponibilidad | ***ID*:**2 - B |
| ***Descripción*:** Este experimento busca verificar la disponibilidad del sistema | ***Responsable*:** Ingenium |
| ***Propósito:***  ( ) Reparación, actualizar, clarificar  ( X ) Obtener Información técnica  ( ) Obtener información de negocio  ( ) Otros: | |
| ***Propósito:***  Verificar las decisiones de diseño asociadas a la disponibilidad del sistema. | |
| ***Descripción del experimento:***  En este experimento debe simular la recepción de información proveniente de los tags RFID que se encuentran en cada uno de los inmuebles. Suponga que por cada inmueble u oficina se tienen en promedio 50 tags RFID enviando información cada segundo a cuatro antenas ubicadas en diferentes posiciones de la casa u oficina. Estas antenas a su vez consolidan la información y la envían a la central. En la central un servidor cuyo objetivo es balancear cargas y verificar la firma digital asociada a cada trama, reparte la carga en dos servidores principales donde se procesa el mensaje y/o evento. | |
| ***Artefactos Creados:***  ASHCentralSystem ->Es el proyecto principal donde se encuentra toda la lógica de negocio para procesar los eventos recibidos por cada casa u oficina.  ASHHomeModule ->Es el proyecto con la lógica para la simulación de los eventos generados por la casa u oficina.  ASHLoadBalancer -> Es el proyecto que contiene la lógica encargada de re direccionar las peticiones recibidas a cada uno de los servidores centrales disponibles, además de manejar la seguridad del sistema. | |
| ***Criterio de terminación:***  Cuando se detecta que un tag RFID ha violado una de las reglas definidas para su comportamiento, se debe generar una alarma indicando a todas las porterías y a los responsables de la vivienda u oficina lo ocurrido. Esto debe suceder en un tiempo inferior a un segundo.  Si un servidor central, deja de estar disponible, los sistemas centrales restantes deben atender la totalidad de las peticiones recibidas sin perder el mensaje.  En caso de que un sistema central fallé y aún no se haya recibido la respuesta de todos los mensajes que fueron enviados hacia él, dichos mensajes deben reenviarse a los servidores centrales restantes disponibles. | |
| ***Recursos Requeridos:***  Recurso humano -> Se necesita el trabajo de 3 personas con conocimientos en java, sockets, estilos y tácticas de arquitectura.  Recursos técnicos ->NetBeans IDE 7.0, Microsoft Visio, JUnit, JUnitPerf | |
| ***Duración estimada:***  18 horas | |

Tabla 55. Resultados del Experimento 2 - B1

|  |
| --- |
| **Resultados del Experimento** |
| ***Resumen de los resultados:***  A continuación se presenta la gráfica de tiempo contra cantidad de usuarios concurrentes. Se puede notar que a medida que se incrementa el número de usuarios concurrentes, el tiempo de respuesta aumenta cada vez más rápido.    En la siguiente gráfica se puede notar como el número de fallas (una falla equivale a que un evento se procesó con un tiempo superior a un segundo) no aumenta hasta llegar a un punto donde se dispara repentinamente (entre 100 y 120 usuarios concurrentes) de manera que en 200 usuarios concurrentes casi el 100% de las peticiones tardan más de un segundo en ser procesadas. Este punto es importante, porque es el que determina la capacidad que tiene el sistema de responder a un número de usuarios concurrentes manteniendo las reglas del negocio (un segundo).    Como puede evidenciarse, con respecto al experimento 1, al incluir nuevos atributos de calidad como seguridad, el tiempo de respuesta aumenta notablemente, pero se está garantizando el procesamiento de todos los mensajes válidos recibidos, a la vez que se ofrece resistencia a un ataque de denegación de servicios y/o spoofing. |
| ***Duración Real:***  14 horas |
| ***Recursos Reales:***  Recurso humano -> Se necesita el trabajo de 2 personas con conocimientos en java, sockets, estilos y tácticas de arquitectura.  Recursos técnicos ->NetBeans IDE 7.0, Microsoft Visio, JUnit, JUnitPerf |
| ***Recomendaciones:***  Para aumentar el desempeño del sistema, se puede mover el módulo de seguridad del servidor de balanceo de cargas a cada uno de los servidores centrales.  Para aumentar disponibilidad, se recomienda aumentar un servidor central al balanceador de cargas. |

* + 1. **Experimento 3 - Interoperabilidad**
       1. **Experimento A - Comunicación con Web Service**

El objetivo de este experimento es comunicar sistemas que funcionen bajo distintas plataformas. De manera más específica, desde el sistema central (Bajo plataforma java) se consumirá un web servicie publicado por un sistema externo (Bajo plataforma .Net). Dicha comunicación se hará a través del esquema XML definido.

Tabla 56. Descripción del Experimento 3 – A

|  |  |
| --- | --- |
| **Descripción del Experimento** | |
| ***Título*:** Interoperabilidad – Comunicación a través de web service entre sistemas. | ***ID*:** A1 |
| ***Descripción*:** Este experimento busca realizar una comunicación entre sistemas desarrollados bajo distintas plataformas (Java y .Net) para favorecer el atributo de calidad Interoperabilidad. | ***Responsable*:** Ingenium |
| ***Propósito:***  ( ) Reparación, actualizar, clarificar  ( X ) Obtener Información técnica  ( ) Obtener información de negocio  ( ) Otros: | |
| ***Propósito:***  Definir un esquema XML para estandarizar la comunicación entre el sistema central y los sistemas externos.  Comunicar sistemas desarrollados bajos distintas tecnologías. | |
| ***Descripción del experimento:***   * Definir un esquema XML para la comunicación entre el sistema central y los sistemas externos. * Construir un Servicio en plataforma .Net que simula el servicio del sistema externo policía. * Desde el sistema central consumir el servicio suministrado por el sistema externo (policía). Dicha comunicación se realizará haciendo uso del esquema XML definido inicialmente. | |
| ***Artefactos Creados:***  ExternalSystem ->Proyecto desarrollado en Microsoft.Net que simulará un sistema externo (policía). Este sistema publicará un WEB service que será consumido por el sistema central.  WEBServiceNotificator -> Módulo en el sistema central que hace llamado al servicio del sistema externo. | |
| ***Criterio de terminación:***  El sistema central debe poder comunicarse con el sistema externo, enviar una notificación y recibir la respuesta del sistema externo haciendo uso del esquema definido inicialmente. | |
| ***Recursos Requeridos:***  Recurso humano -> Se necesita el trabajo de 2 personas con conocimientos en java, JEE, JSF, Microsoft.Net y web services.  Recursos técnicos -> NetBeans IDE 7.0, Microsoft VS.Net 2008. | |
| ***Duración estimada:***  10 horas | |

Tabla 57. Resultados del Experimento 3 -A1

|  |
| --- |
| **Resultados del Experimento** |
| ***Definición de esquema xml:***  ***Esquema 1:*** Este esquema xml permite al sistema central informar a un sistema externo acerca de la generación de un evento que solicita la atención por parte del sistema externo.  Homeid: Identificador de la casa que generó el evento.  EventType: Tipo de evento generado en la casa  Time: Fecha y hora de generación del evento  ContactName: Nombre del contacto asociado a la casa (dueño y/o habitante)  CantactPhone: Teléfono donde se puede ubicar al dueño y/o habitante de la casa.  <?xmlversion="1.0"encoding="utf-8"?>  <xs:schemaid="XMLNotificationSchema"  targetNamespace="http://tempuri.org/XMLNotificationSchema.xsd"  elementFormDefault="qualified"  xmlns="http://tempuri.org/XMLNotificationSchema.xsd"  xmlns:mstns="http://tempuri.org/XMLNotificationSchema.xsd"  xmlns:xs="http://www.w3.org/2001/XMLSchema"  >  <xs:elementname="notification">  <xs:complexType>  <xs:sequence>  <xs:elementname="homeid" />  <xs:elementname="eventtype"></xs:element>  <xs:elementname="time"></xs:element>  <xs:elementname="contactname"></xs:element>  <xs:elementname="contactphone"></xs:element>  <xs:elementname="address"></xs:element>  <xs:elementname="city"></xs:element>  </xs:sequence>  </xs:complexType>  </xs:element>  </xs:schema>  ***Esquema 2:*** *Con este esquema xml, el sistema externo genera una respuesta para el sistema central, respuesta que confirma que la notificación se realizó correctamente.*  *Homeid: Identificador de la casa de la cual se recibió la notificación.*  *EventType: Evento que fue notificado.*  *Message: Mensaje de respuesta generado por el sistema externo, donde informa el tiempo aproximado de llegada al lugar de los hechos.*  <?xmlversion="1.0"encoding="utf-8"?>  <xs:schemaid="XMLResponseSchema"  targetNamespace="http://tempuri.org/XMLResponseSchema.xsd"  elementFormDefault="qualified"  xmlns="http://tempuri.org/XMLResponseSchema.xsd"  xmlns:mstns="http://tempuri.org/XMLResponseSchema.xsd"  xmlns:xs="http://www.w3.org/2001/XMLSchema"  >  <xs:elementname="response">  <xs:complexType>  <xs:sequence>  <xs:elementname="homeid" />  <xs:elementname="eventtype"></xs:element>  <xs:elementname="message"></xs:element>  </xs:sequence>  </xs:complexType>  </xs:element>  </xs:schema>  ***WSDL Asociado:*** *A continuación se presenta el wsdl publicado por el sistema externo que permite la comunicación entre los dos sistemas.*  <?xml version="1.0" encoding="utf-8" ?>  [**-**](http://localhost/ExternalSystemPolice/PoliceService.asmx?wsdl)<wsdl:definitionsxmlns:soap="**http://schemas.xmlsoap.org/wsdl/soap/**"xmlns:tm="**http://microsoft.com/wsdl/mime/textMatching/**"xmlns:soapenc="**http://schemas.xmlsoap.org/soap/encoding/**"xmlns:mime="**http://schemas.xmlsoap.org/wsdl/mime/**"xmlns:tns="**http://tempuri.org/**"xmlns:s="**http://www.w3.org/2001/XMLSchema**" xmlns:soap12="**http://schemas.xmlsoap.org/wsdl/soap12/**"xmlns:http="**http://schemas.xmlsoap.org/wsdl/http/**"targetNamespace="**http://tempuri.org/**"xmlns:wsdl="**http://schemas.xmlsoap.org/wsdl/**">  [**-**](http://localhost/ExternalSystemPolice/PoliceService.asmx?wsdl)<wsdl:types>  [**-**](http://localhost/ExternalSystemPolice/PoliceService.asmx?wsdl)<s:schemaelementFormDefault="**qualified**"targetNamespace="**http://tempuri.org/**">  [**-**](http://localhost/ExternalSystemPolice/PoliceService.asmx?wsdl)<s:element name="**HelloWorld**">   <s:complexType/>   </s:element>  [**-**](http://localhost/ExternalSystemPolice/PoliceService.asmx?wsdl)<s:element name="**HelloWorldResponse**">  [**-**](http://localhost/ExternalSystemPolice/PoliceService.asmx?wsdl)<s:complexType>  [**-**](http://localhost/ExternalSystemPolice/PoliceService.asmx?wsdl)<s:sequence>   <s:elementminOccurs="**0**"maxOccurs="**1**" name="**HelloWorldResult**" type="**s:string**" />   </s:sequence>   </s:complexType>   </s:element>  [**-**](http://localhost/ExternalSystemPolice/PoliceService.asmx?wsdl)<s:element name="**NotifyEvent**">  [**-**](http://localhost/ExternalSystemPolice/PoliceService.asmx?wsdl)<s:complexType>  [**-**](http://localhost/ExternalSystemPolice/PoliceService.asmx?wsdl)<s:sequence>   <s:elementminOccurs="**0**"maxOccurs="**1**" name="**xmlEventNotification**" type="**s:string**" />   </s:sequence>   </s:complexType>   </s:element>  [**-**](http://localhost/ExternalSystemPolice/PoliceService.asmx?wsdl)<s:element name="**NotifyEventResponse**">  [**-**](http://localhost/ExternalSystemPolice/PoliceService.asmx?wsdl)<s:complexType>  [**-**](http://localhost/ExternalSystemPolice/PoliceService.asmx?wsdl)<s:sequence>   <s:elementminOccurs="**0**"maxOccurs="**1**" name="**NotifyEventResult**" type="**s:string**" />   </s:sequence>   </s:complexType>   </s:element>   </s:schema>   </wsdl:types>  [**-**](http://localhost/ExternalSystemPolice/PoliceService.asmx?wsdl)<wsdl:message name="**HelloWorldSoapIn**">   <wsdl:partname="**parameters**" element="**tns:HelloWorld**" />   </wsdl:message>  [**-**](http://localhost/ExternalSystemPolice/PoliceService.asmx?wsdl)<wsdl:message name="**HelloWorldSoapOut**">   <wsdl:partname="**parameters**" element="**tns:HelloWorldResponse**" />   </wsdl:message>  [**-**](http://localhost/ExternalSystemPolice/PoliceService.asmx?wsdl)<wsdl:message name="**NotifyEventSoapIn**">   <wsdl:partname="**parameters**" element="**tns:NotifyEvent**" />   </wsdl:message>  [**-**](http://localhost/ExternalSystemPolice/PoliceService.asmx?wsdl)<wsdl:message name="**NotifyEventSoapOut**">   <wsdl:partname="**parameters**" element="**tns:NotifyEventResponse**" />   </wsdl:message>  [**-**](http://localhost/ExternalSystemPolice/PoliceService.asmx?wsdl)<wsdl:portType name="**PoliceServiceSoap**">  [**-**](http://localhost/ExternalSystemPolice/PoliceService.asmx?wsdl)<wsdl:operation name="**HelloWorld**">   <wsdl:inputmessage="**tns:HelloWorldSoapIn**" />   <wsdl:outputmessage="**tns:HelloWorldSoapOut**" />   </wsdl:operation>  [**-**](http://localhost/ExternalSystemPolice/PoliceService.asmx?wsdl)<wsdl:operation name="**NotifyEvent**">   <wsdl:inputmessage="**tns:NotifyEventSoapIn**" />   <wsdl:outputmessage="**tns:NotifyEventSoapOut**" />   </wsdl:operation>   </wsdl:portType>  [**-**](http://localhost/ExternalSystemPolice/PoliceService.asmx?wsdl)<wsdl:binding name="**PoliceServiceSoap**" type="**tns:PoliceServiceSoap**">   <soap:bindingtransport="**http://schemas.xmlsoap.org/soap/http**" />  [**-**](http://localhost/ExternalSystemPolice/PoliceService.asmx?wsdl)<wsdl:operation name="**HelloWorld**">   <soap:operationsoapAction="**http://tempuri.org/HelloWorld**" style="**document**" />  [**-**](http://localhost/ExternalSystemPolice/PoliceService.asmx?wsdl)<wsdl:input>   <soap:bodyuse="**literal**" />   </wsdl:input>  [**-**](http://localhost/ExternalSystemPolice/PoliceService.asmx?wsdl)<wsdl:output>   <soap:bodyuse="**literal**" />   </wsdl:output>   </wsdl:operation>  [**-**](http://localhost/ExternalSystemPolice/PoliceService.asmx?wsdl)<wsdl:operation name="**NotifyEvent**">   <soap:operationsoapAction="**http://tempuri.org/NotifyEvent**" style="**document**" />  [**-**](http://localhost/ExternalSystemPolice/PoliceService.asmx?wsdl)<wsdl:input>   <soap:bodyuse="**literal**" />   </wsdl:input>  [**-**](http://localhost/ExternalSystemPolice/PoliceService.asmx?wsdl)<wsdl:output>   <soap:bodyuse="**literal**" />   </wsdl:output>   </wsdl:operation>   </wsdl:binding>  [**-**](http://localhost/ExternalSystemPolice/PoliceService.asmx?wsdl)<wsdl:binding name="**PoliceServiceSoap12**" type="**tns:PoliceServiceSoap**">   <soap12:bindingtransport="**http://schemas.xmlsoap.org/soap/http**" />  [**-**](http://localhost/ExternalSystemPolice/PoliceService.asmx?wsdl)<wsdl:operation name="**HelloWorld**">   <soap12:operationsoapAction="**http://tempuri.org/HelloWorld**" style="**document**" />  [**-**](http://localhost/ExternalSystemPolice/PoliceService.asmx?wsdl)<wsdl:input>   <soap12:bodyuse="**literal**" />   </wsdl:input>  [**-**](http://localhost/ExternalSystemPolice/PoliceService.asmx?wsdl)<wsdl:output>   <soap12:bodyuse="**literal**" />   </wsdl:output>   </wsdl:operation>  [**-**](http://localhost/ExternalSystemPolice/PoliceService.asmx?wsdl)<wsdl:operation name="**NotifyEvent**">   <soap12:operationsoapAction="**http://tempuri.org/NotifyEvent**" style="**document**" />  [**-**](http://localhost/ExternalSystemPolice/PoliceService.asmx?wsdl)<wsdl:input>   <soap12:bodyuse="**literal**" />   </wsdl:input>  [**-**](http://localhost/ExternalSystemPolice/PoliceService.asmx?wsdl)<wsdl:output>   <soap12:bodyuse="**literal**" />   </wsdl:output>   </wsdl:operation>   </wsdl:binding>  [**-**](http://localhost/ExternalSystemPolice/PoliceService.asmx?wsdl)<wsdl:service name="**PoliceService**">  [**-**](http://localhost/ExternalSystemPolice/PoliceService.asmx?wsdl)<wsdl:port name="**PoliceServiceSoap**" binding="**tns:PoliceServiceSoap**">   <soap:addresslocation="**http://localhost/ExternalSystemPolice/PoliceService.asmx**" />   </wsdl:port>  [**-**](http://localhost/ExternalSystemPolice/PoliceService.asmx?wsdl)<wsdl:port name="**PoliceServiceSoap12**" binding="**tns:PoliceServiceSoap12**">   <soap12:addresslocation="**http://localhost/ExternalSystemPolice/PoliceService.asmx**" />   </wsdl:port>   </wsdl:service>   </wsdl:definitions>  El sistema central al recibir un evento consume el servicio Web invocando a “NotifyEvent” y pasándole como parámetro una cadena en formato XML cumpliendo con el esquema 1. El sistema externo recibe dicha notificación, genera un mensaje de respuesta cumpliendo con el esquema 2 y lo retorna al invocador (sistema central). Por último el sistema central, presenta el mensaje proveniente del sistema externo.  El sistema externo simula un tiempo de respuesta lento. Aquí se hace uso de comunicación asíncrona, para que el sistema central consuma el servicio web por cada evento que se genera. De esta manera se envían notificaciones a los sistemas externos sin tener que esperar la respuesta de dicho sistema (asíncrono). Esto favorece en gran medida el tiempo de desempeño del sistema central.  La definición de un esquema XML para la comunicación de sistemas, favorece la interoperabilidad. Sin importar la tecnología que ha implementado el sistema externo, la comunicación se hace de manera transparente. Y si llegado el caso se requiere adicionar un nuevo parámetro para la comunicación, basta con modificar el esquema de comunicación, pero el servicio no requeriría modificación alguna.  La comunicación desde el sistema central, se realiza a través de un proxy generado de acuerdo al wsdl que publica el sistema externo. En caso de que el sistema externo adicione un nuevo servicio y éste requiera ser consumido por el sistema central, el proxy debe ser modificado / actualizado. |
| ***Duración Real:***  10 horas |
| ***Recursos Reales:***  Recurso humano -> Se necesita el trabajo de 2 personas con conocimientos en java, JEE, JSF, Microsoft.Net y web services.  Recursos técnicos ->NetBeans IDE 7.0, Microsoft VS.Net 2008. |
| ***Recomendaciones:***  Desde el sistema central se debe implementar un mecanismo de generación del proxy dinámico. Esto debido a que el web service suministrado por el sistema externo puede cambiar su estructura y/o agregar nuevos servicios. |

* + - 1. **Experimento B - Comunicación Asincrona – Envío de correo electrónico**

Tabla 58. Descripción del Experimento 3 - B

|  |  |
| --- | --- |
| **Descripción del Experimento** | |
| ***Título*:**Interoperabilidad comunicación asíncrona | ***ID*: 3 -** B |
| ***Descripción*:**  Se deben comunicar los eventos de manera asíncrona haciendo uso del correo electrónico. | ***Responsable*:**Ingenium |
| ***Propósito:***  ( ) Reparación, actualizar, clarificar  ( X ) Obtener Información técnica  ( X ) Obtener información de negocio  ( ) Otros: | |
| ***Propósito:***  Verificar y validar los artefactos desarrollados en la arquitectura planteada de manera que sea posible la interoperabilidad para la comunicación de los eventos a los clientes de manera asíncrona. | |
| ***Descripción del experimento:***  Verificar la arquitectura para garantizar la interoperabilidad del sistema, de manera que se pueda comunicar de forma asíncrona con los operadores de correo en internet, con el fin de informar vía correo electrónico un evento pre configurado por el cliente.  Se integra a los desarrollos del sistema central un nuevo módulo encargado de realizar la notificación a los clientes de los eventos detectados a través de correo electrónico. De manera que se validen las decisiones de diseño asociadas a la interoperabilidad del sistema, cumpliendo con las expectativas planteadas. | |
| ***Artefactos Creados:***  ASHCentralSystem / Email Notificator ->Es un módulo del sistema central en la capa de notificación encargado de enviar las notificaciones de manera asíncrona a través a los operadores de correo electrónico. | |
| ***Criterio de terminación:***  El sistema central recibe un evento y realiza el envió de un correo electrónico al cliente a través de un servidor de correo en internet, informando del evento que ha sucedido den su hogar u oficina. | |
| ***Recursos Requeridos:***  Recurso humano -> Se necesita el trabajo de 2 personas con conocimientos en java, estilos y tácticas de arquitectura, bases de datos, Threads, etc.  Recursos técnicos -> Base de datos Oracle XE, NetBeans IDE 7.0, Microsoft Visio, JUnit, Cuenta de correo electrónico. | |
| ***Duración estimada:***  10 horas | |

Tabla 59. Resultados del Experimento 3 - B

|  |
| --- |
| **Resultados del Experimento** |
| ***Resumen de los resultados:***  Se agregó a la capa de notificación dentro del sistema central un nuevo módulo encargado de las notificaciones a través de correo electrónico, no se debió modificar ninguna otra parte del sistema, se descubrió que se deben validar los casos en los que se debe enviar notificación al cliente, haciendo uso de esta herramienta.  Adicionalmente se definió un formato de mensaje que permitiera estandarizar las comunicaciones a los clientes y permitir el uso del módulo EmailNotificator para notificar cualquier tipo de evento.  El mensaje recibido por un cliente tiene el siguiente aspecto.    En el sistema central se presenta un patrón arquitectural por capas, y adicionalmente se tiene módulos con responsabilidades muy claras, permitiendo bajo acoplamiento y alta extensibilidad, adicionalmente permite favorecer la interoperabilidad. De esta manera se garantizó que se cumplieran con las expectativas de los interesados, de manera específica agregando el nuevo tipo de modificación al sistema. |
| ***Duración Real:***  12 horas |
| ***Recursos Reales:***  Recurso humano -> Se necesita el trabajo de 2 personas con conocimientos en java, estilos y tácticas de arquitectura, bases de datos, Threads, etc.  Recursos técnicos -> Base de datos Oracle XE, NetBeans IDE 7.0, Microsoft Visio, JUnit, Cuenta de correo electrónico. |
| ***Recomendaciones:***  Se recomienda verificar las restricciones de negocio para hacer uso de cada uno de los tipos de notificación en el momento indicado.  Se debe realizar la notificación por correo al cliente en un hilo de ejecución diferente al hilo de ejecución de notificación de los sistemas de emergencia, o se debe realizar la notificación después de hacer la notificación a los sistemas de emergencia para no ir en contra de las restricciones ya definidas y los demás atributos de calidad.  Se debe seguir haciendo uso de la plantilla definida, permitiendo integrar cualquier tipo de notificación al módulo ya creado. |

* + - 1. **Experimento C - Video llamada**

Tabla 60. Descripción del Experimento 3 - C

|  |  |
| --- | --- |
| **Descripción del Experimento** | |
| ***Título*:** Video llamada de emergencia | ***ID*:**3 - C |
| ***Descripción*:** Este experimento busca conseguir realizar una video llamada de emergencia como respuesta a un evento | ***Responsable*:**Ingenium |
| ***Propósito:***  ( ) Reparación, actualizar, clarificar  ( ) Obtener Información técnica  ( ) Obtener información de negocio  ( X ) Otros: | |
| ***Propósito:***   * Notificar a los suscriptores sobre la detección de un evento para el cual han sido previamente suscritos * Iniciar automáticamente las video llamadas de las cámaras. * Reportar el estado de las llamadas | |
| ***Descripción del experimento:***  Hacer uso del API de Skype para lograr que nuestra aplicación genere una video llamada desde un equipo (PC) que se encuentre instalado en el domicilio y que cuente con una cámara Web disponible y correctamente configurada. También se incluyó la funcionalidad necesaria para que en caso de que la video llamada sea fallida, se envíe un correo de notificación al usuario suscrito, informándole sobre el evento ocurrido; por ejemplo en el caso de que no sea contestada, o que el usuario al que se esté llamando no se encuentre disponible. | |
| ***Artefactos Creados:***  Se ha modificado el proyecto ASHHomeModule para que incluya las funciones necesarias para la realización de la video llamada desde el computador que se encuentra instalado en el hogar, así colas funciones necesarias para el envío del correo electrónico informativo en caso de que la llamada sea fallida | |
| ***Criterio de terminación:***  El equipo instalado en el hogar debe ser capaz de generar la llamada a través de Skype automáticamente, e iniciar la transmisión de video en caso de que la llamada sea contestada o enviar un correo de notificación en caso contrario. | |
| ***Recursos Requeridos:***  Recurso humano -> Se necesita el trabajo de 2 personas con conocimientos generales sobre desarrollo de software para el API de Skype y específicamente en Java para el desarrollo de la aplicación  Recurso técnico ->NetBeans IDE 7.0, Referencia de la documentación de la API de Skype. Cuenta de Skype activa. Herramienta de consola para Skype (Tracer.exe) | |
| ***Duración estimada:***  8 horas | |

Tabla 61. Resultados del Experimento 3 - C

|  |
| --- |
| **Resultados del Experimento** |
| ***Resumen de los resultados:***  Se revisó la documentación de la API de Skype con el fin de encontrar las instrucciones y comandos necesarios para la generación de llamadas. Una vez verificadas las instrucciones necesarias se procedió a realizar las llamadas a través de la consola de Skype por medio de la herramienta **Tracer.exe.** Por medio del uso de esta herramienta se pudo determinar el funcionamiento de las llamadas de Skype.  Este es un ejemplo de la consola de Skype    Uno de los requisitos para poder realizar la llamada desde Skype es que el equipo desde donde se realizara la llamada cuente con una cuenta activa de Skype y que se encuentre logeado en la misma. Si se cumple este requisito se puede generar una llamada de voz. Si la llamada de voz es respondida, se genera un mensaje acerca del estado de la llamada y se devuelve un identificador único de la llamada. Con este identificador se puede proceder a realizar la modificación de la llamada, para nuestro caso, iniciar la transmisión de video. Por medio de esta experimentación se pudo determinar que no es posible iniciar la llamada directamente con video puesto que el video es una característica de una llamada de voz que solo puede ser habilitada una vez la llamada inicial sea exitosa.  Con estos resultados, se procedió a crear un proyecto de prueba en Java para la generación de la llamada desde la aplicación basándose en otros ejemplos los cuales usan un framework de Skype que encapsula las funcionalidades de la API de Skype para su uso en Java de una manera más amigable y entendible. Sobre este proyecto de prueba se realizó la experimentación necesaria para verificar que se podía generar correctamente la llamada e iniciar el video de la misma desde la aplicación. Igualmente se realizó la implementación necesaria para enviar un mensaje en el chat de Skype informando previamente al usuario que se realizara una video llamada para que el usuario la conteste.  Una vez lograda los objetivos de la generación de la video llamada, se procedió a incluir la codificación generada en el proyecto de prueba en el proyecto **ASHHomeModule** previamente creado para soportar las funcionalidades de llamada y video llamada en Skype. También se modificó el proyecto para que en caso de que la llamada sea fallida, se envíe un correo de notificación al usuario para que tome las medidas correspondientes según el evento que se registre. |
| ***Duración Real:***  12 horas |
| ***Recursos Reales:***  Recurso humano -> Se necesita el trabajo de 2 personas con conocimientos en java, sockets, estilos y tácticas de arquitectura, así como el análisis de la documentación del API de Skype  Recursos técnicos ->NetBeans IDE 7.0, Documentación de Skype, cuenta de Skype activa, cuenta de correo activa, consola de Skype, Framework de Skype, librerías de Skype. |
| ***Recomendaciones:***  Uno de los limitantes de usar una video llamada de Skype es que se debe tener un computador encendido con una cuenta de Skype activa y logeada antes de poder iniciar la video llamada, esto conlleva al hecho que si el equipo se encuentra fuera de línea (apagado o sin red o sin logear) la llamada no va ser exitosa, lo cual genera una de deficiencia pues cualquiera de los actores del sistema puede deshabilitarlo intencional o accidentalmente.  Se pueden conectar diferentes cámaras a un mismo sistema, pero estas deben contar con identificadores únicos para poder iniciar la transmisión de video desde una de ellas, la limitante es que solo se puede transmitir desde una de ellas en un mismo instante de tiempo, es decir si se generan alertas en diferentes áreas del domicilio solo se generara la llamada del primer evento con la cámara registrada a ese evento. El resto de los eventos no podrán generar la llamada porque la cuenta activa ya se encuentra ocupada. |

1. **Bibliografía**

* http://es.wikipedia.org/wiki/Aplicación\_web
* Proyecto – Casa Intelingente, Uniandes, 2011
* Documento de Arquitectura del Sistema (SAD), Uniandes, 2011
* ACDM – Roles, Uniandes, 2011
* Documentación de la API de Skype: <http://developer.skype.com/resources/public_api_ref.zip>
* Herramienta de consola de Skype: <http://developer.skype.com/resources/Tracer.exe>
* Ejemplos de desarrollo : <https://github.com/taksan/skype-java-api>

<http://skype.sourceforge.jp/index.php?Skype%20API%20For%20Java%20%28English%29>