**Especificaciones de Requisitos de Software**

**Para**

**SMART HOME SYSTEM**

Junio 2019

|  |  |
| --- | --- |
| Atributos | Contenido |
| Cliente: | Uriel Calderón Uribe |
| Nombre Proyecto: | Smart Home System |
| Título del Documento: | Especificación de requerimientos de software |
| Fecha: | 28 de junio del 2019 |

**Histórico de cambios:**

|  |  |  |  |
| --- | --- | --- | --- |
| Versión | Fecha de Revisión | Modificado por | Descripción |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

# Introducción

## El IoT (Internet of Things, Internet de las Cosas) se refiere a una red de objetos cotidianos interconectados con representaciones virtuales en una estructura similar al internet, de hecho, se considera una extensión de este que permite la conexión entre los objetos físicos y los dispositivos. Hoy en día una gran parte de electrodomésticos pueden ser configurados para trabajar en un entorno de IoT, por lo que su implementación en los hogares cada vez es más frecuente para aquellas personas que tiene un estilo de vida muy activo o que simplemente su agenda no les permite estar mucho en casa, ya que les permite estar conectados con su hogar y consultar y controlar actividades e información de este.

## Propósito

## El presente documento tiene como propósito definir los objetivos del proyecto Smart Home System la justificación de este, los interesados en él, todo ello para la implementación de un sistema que permitirá administrar procesos del hogar mediante una aplicación móvil.

## Alcance

El presente proyecto se centra en implementar un sistema Smart House para los clientes interesados, dicho sistema tiene que ser simple, eficaz, que automatice tareas del hogar como lo es el control de la iluminación, aire acondicionado, alarma, etc. En otras palabras, crear un sistema de calidad ajustado a las necesidades del cliente.

## Personal involucrado

|  |  |
| --- | --- |
| Nombre | Bernardo Quintino Guzmán |
| Rol | Jefe y desarrollador de proyecto Smart House |
| Categoría profesional | Ingeniero en sistemas de computo |
| Responsabilidades | Administrar tareas y actividades del proyecto. |
| Información de contacto | b.quininoguzman@ugto.mx |

|  |  |
| --- | --- |
| Nombre | Eliza Guadalupe Méndez Zavala |
| Rol | Desarrolladora de modelos Smart House |
| Categoría profesional | Ingeniero en sistemas de computo |
| Responsabilidades | Diseñar los modelos a escala para el proyecto Smart House |
| Información de contacto | eg.mendezzavala@ugto.mx |

|  |  |
| --- | --- |
| Nombre | Martin García Hurtado |
| Rol | Desarrollador de modelos Smart House |
| Categoría profesional | Ingeniero en sistemas de computo |
| Responsabilidades | Diseñar los modelos a escala para el proyecto Smart House |
| Información de contacto | m.garciahurtado@ugto.mx |

|  |  |
| --- | --- |
| Nombre | Francisco David Gámez Pérez |
| Rol | Desarrollador de software |
| Categoría profesional | Ingeniero en sistemas de computo |
| Responsabilidades | Desarrollar software para el proyecto Smart House |
| Información de contacto | Fd.gamezperez@ugto.mx |

|  |  |
| --- | --- |
| Nombre | Cristian Almanza Armenta |
| Rol | Desarrollador de software |
| Categoría profesional | Ingeniero en sistemas de computo |
| Responsabilidades | Desarrollar software para el proyecto Smart House |
| Información de contacto | c.almanzaarmenta@ugto.mx |

## **Definiciones, acrónimos y abreviaturas**

De tecnología:

1. **Arduino:** Es una plataforma de hardware libre, basada en una placa con un microcontrolador y un entorno de desarrollo, diseñado para facilitar el uso de la electrónica aplicada en diferentes entornos.
2. **Sensor:** Dispositivo que capta magnitudes físicas (variaciones de luz, temperatura, sonido, etc.) u otros presentes en el entorno.
3. **Programación:** Instrumento que permite la ejecución de tareas en un sistema informático.
4. **Programa:** Conjunto de instrucciones para el ordenador que permite la ejecución de tareas o instrucciones definidas y ordenadas.
5. **Automatización:** La automatización es el conjunto de elementos o procesos informáticos, mecánicos y electromecánicos que operan con mínima o nula intervención del ser humano.
6. **Aplicación:** Es un programa que puede ser instalado en dispositivos móviles y computadoras para el usuario, para la realización de una o varias tareas.
7. **Android:** Sistema operativo que se emplea en dispositivos móviles de pantalla táctil.
8. **Modelo a escala:** Representación de un objeto en diferente tamaño al original.

## Referencias

|  |  |  |  |
| --- | --- | --- | --- |
| **Referencia** | **Titulo** | **Fecha** | **Autor** |
| Libro | El Proceso Unificado de Desarrollo de Software | 2000 | JACOBSON, Ivar; BOOCH, Grady; RUMBAUGH, James |
| Libro | Ingeniería del Software, un enfoque Práctico (Quinta edición edición) | 2003 | Pressman, Roger S. |
| Libro | Formalización del proceso de desarrollo de software | 1998 | Haeberer, A. M.; P. A. S. Veloso, G. Baum |
| Libro | System Requirements Engineering | 1995 | Loucopoulos, Pericles; Karakostas, V. |

## Resumen

El presente documento presenta las especificaciones de requisitos para el proyecto Smart House System, dichas especificaciones contemplan distintos aspectos de los requisitos como la funcionalidad del producto, interfaz de usuario, requisitos funcionales, no funcionales, así como los tipos de usuario contemplados en el sistema y la seguridad del mismo.

# Descripción general

## Perspectiva del producto

Indicar si es un producto independiente o parte de un sistema mayor. En el caso de tratarse de un producto que forma parte de un sistema mayor, un diagrama que sitúe el producto dentro del sistema e identifique sus conexiones facilita la comprensión.

## Funcionalidad del producto

Resumen de las funcionalidades principales que el producto debe realizar, sin entrar en información de detalle.

En ocasiones la información de esta sección puede tomarse de un documento de especificación del sistema de mayor nivel (ej. Requisitos del sistema).

Las funcionalidades deben estar organizadas de manera que el cliente o cualquier interlocutor pueda entenderlo perfectamente. Para ello se pueden utilizar métodos textuales o gráficos.

## Características de los usuarios

|  |  |
| --- | --- |
| Tipo de usuario |  |
| Formación |  |
| Habilidades |  |
| Actividades |  |

Descripción de los usuarios del producto, incluyendo nivel educacional, experiencia y experiencia técnica.

## Restricciones

Descripción de aquellas limitaciones a tener en cuenta a la hora de diseñar y desarrollar el sistema, tales como el empleo de determinadas metodologías de desarrollo, lenguajes de programación, normas particulares, restricciones de hardware, de sistema operativo etc.

## Suposiciones y dependencias

Descripción de aquellos factores que, si cambian, pueden afectar a los requisitos. Por ejemplo una asunción puede ser que determinado sistema operativo está disponible para el hardware requerido. De hecho, si el sistema operativo no estuviera disponible, la SRS debería modificarse.

## Evolución previsible del sistema

Identificación de futuras mejoras al sistema, que podrán analizarse e implementarse en un futuro.

# Requisitos específicos

Esta es la sección más extensa y más importante del documento.

Debe contener una lista detallada y completa de los requisitos que debe cumplir el sistema a desarrollar. El nivel de detalle de los requisitos debe ser el suficiente para que el equipo de desarrollo pueda diseñar un sistema que satisfaga los requisitos y los encargados de las pruebas puedan determinar si éstos se satisfacen.

Los requisitos se dispondrán en forma de listas numeradas para su identificación, seguimiento, trazabilidad y validación (ej. RF 10, RF 10.1, RF 10.2,...).

Para cada requisito debe completarse la siguiente tabla:

|  |  |  |  |
| --- | --- | --- | --- |
| Número de requisito |  | | |
| Nombre de requisito |  | | |
| Tipo | Requisito | Restricción | |
| Fuente del requisito |  | | |
| Prioridad del requisito | Alta/Esencial | Media/Deseado | Baja/ Opcional |

y realizar a continuación la descripción del requisito

La distribución de los párrafos que forman este punto puede diferir del propuesto en esta plantilla, si las características del sistema aconsejan otra distribución para ofrecer mayor claridad en la exposición.

## Requisitos comunes de los interfaces

Descripción detallada de todas las entradas y salidas del sistema de software.

### Interfaces de usuario

Describir los requisitos del interfaz de usuario para el producto. Esto puede estar en la forma de descripciones del texto o pantallas del interfaz. Por ejemplo posiblemente el cliente ha especificado el estilo y los colores del producto. Describa exacto cómo el producto aparecerá a su usuario previsto.

### Interfaces de hardware

Especificar las características lógicas para cada interfaz entre el producto y los componentes de hardware del sistema. Se incluirán características de configuración.

### Interfaces de software

Indicar si hay que integrar el producto con otros productos de software.

Para cada producto de software debe especificarse lo siguiente:

* Descripción del producto software utilizado
* Propósito del interfaz
* Definición del interfaz: contiendo y formato

### Interfaces de comunicación

Describir los requisitos del interfaces de comunicación si hay comunicaciones con otros sistemas y cuales son las protocolos de comunicación.

## Requisitos funcionales

Definición de acciones fundamentales que debe realizar el software al recibir información, procesarla y producir resultados.

En ellas se incluye:

* Comprobación de validez de las entradas
* Secuencia exacta de operaciones
* Respuesta a situaciones anormales (desbordamientos, comunicaciones, recuperación de errores)
* Parámetros
* Generación de salidas
* Relaciones entre entradas y salidas (secuencias de entradas y salidas, formulas para la conversión de información)
* Especificación de los requisitos lógicos para la información que será almacenada en base de datos (tipo de información, requerido)

Las requisitos funcionales pueden ser divididos en sub-secciones.

### Requisito funcional 1

### Requisito funcional 2

### Requisito funcional 3

### Requisito funcional n

## Requisitos no funcionales

### Requisitos de rendimiento

Especificación de los requisitos relacionados con la carga que se espera tenga que soportar el sistema. Por ejemplo, el número de terminales, el número esperado de usuarios simultáneamente conectados, número de transacciones por segundo que deberá soportar el sistema, etc.

Todos estos requisitos deben ser mesurables. Por ejemplo, indicando “el 95% de las transacciones deben realizarse en menos de 1 segundo”, en lugar de “los operadores no deben esperar a que se complete la transacción”.

### Seguridad

Especificación de elementos que protegerán al software de accesos, usos y sabotajes maliciosos, así como de modificaciones o destrucciones maliciosas o accidentales. Los requisitos pueden especificar:

* Empleo de técnicas criptográficas.
* Registro de ficheros con “logs” de actividad.
* Asignación de determinadas funcionalidades a determinados módulos.
* Restricciones de comunicación entre determinados módulos.
* Comprobaciones de integridad de información crítica.

### Fiabilidad

Especificación de los factores de fiabilidad necesaria del sistema. Esto se expresa generalmente como el tiempo entre los incidentes permisibles, o el total de incidentes permisible.

### Disponibilidad

Especificación de los factores de disponibilidad final exigidos al sistema. Normalmente expresados en % de tiempo en los que el software tiene que mostrar disponibilidad.

### Mantenibilidad

Identificación del tipo de mantenimiento necesario del sistema.

Especificación de quien debe realizar las tareas de mantenimiento, por ejemplo usuarios, o un desarrollador.

Especificación de cuando debe realizarse las tareas de mantenimiento. Por ejemplo, generación de estadísticas de acceso semanales y mensuales.

### Portabilidad

Especificación de atributos que debe presentar el software para facilitar su traslado a otras plataformas u entornos. Pueden incluirse:

* Porcentaje de componentes dependientes del servidor.
* Porcentaje de código dependiente del servidor.
* Uso de un determinado lenguaje por su portabilidad.
* Uso de un determinado compilador o plataforma de desarrollo.
* Uso de un determinado sistema operativo.

## Otros requisitos

Cualquier otro requisito que no encaje en ninguna de las secciones anteriores.

Por ejemplo:

Requisitos culturales y políticos

Requisitos Legales

# Apéndices

Pueden contener todo tipo de información relevante para la SRS pero que, propiamente, no forme parte de la SRS.

# Introducción

## Propósito

## Alcance

## Personal involucrado

## Definiciones, acrónimos y abreviaturas

## Referencias

## Resumen

# Descripción general

## Perspectiva del producto

## Funcionalidad del producto

## Características de los usuarios

## Restricciones

## Suposiciones y dependencias

## Evolución previsible del sistema

# Requisitos específicos

## Requisitos comunes de los interfaces

## itos funcionales

## Requisitos no funcionales

## Otros requisitos

# Apéndices