

Documento de Requerimientos

Aplicación DomoLinx

Versión:

1.0 elaborada por el grupo Linces.

1.1 elaborada por el grupo Linces

Fecha:

25/06/2017.

Índice

Introducción.....	3
Glosario.....	3
Requerimientos de usuario.....	3
Diagramas de UML.....	4
Arquitectura del sistema.....	6
Especificación de los requerimientos del sistema.....	6
Matriz de trazabilidad.....	6
Diseño de casos de prueba del sistema.....	7

Introducción

El siguiente documento de requerimientos sobre el software DomoLinx será desarrollado tanto con el cliente en mente como también así con los desarrolladores de la aplicación.

La aplicación DomoLinx es necesaria para ubicar al grupo Lince como un competente grupo de desarrolladores de aplicaciones apuntadas a la solución de problemas de domótica.

DomoLinx ofrece varias funcionalidades. La aplicación ofrece una interfaz gráfica con representación de datos recolectados por un sensor de humedad y temperatura en tiempo real y controlar dispositivos conectados al sistema como un aire acondicionado, una estufa eléctrica y un humidificador de ambiente. Al usuario se le ofrecen tres opciones para el monitoreo y control del sistema.

1. **Modo monitor**: Permite al usuario monitorear en tiempo real el estado del sistema, los datos recolectados por el sensor de temperatura y humedad y el estado de los periféricos conectados al sistema.
2. **Modo preset**: Al usuario se le ofrece establecer valores de temperatura y humedad en base a los cuales la aplicación tomará decisiones de control sobre los dispositivos manejados.
3. **Generación de reporte**: Se genera un reporte del estado instantáneo del sistema para archivar.

Glosario

- **SRS**: Software requirements specification o especificación de requerimientos del software
- **AC**: Air conditioning o aire acondicionado.
- **MVC**: Model view controller o modelo-vista-controlador.

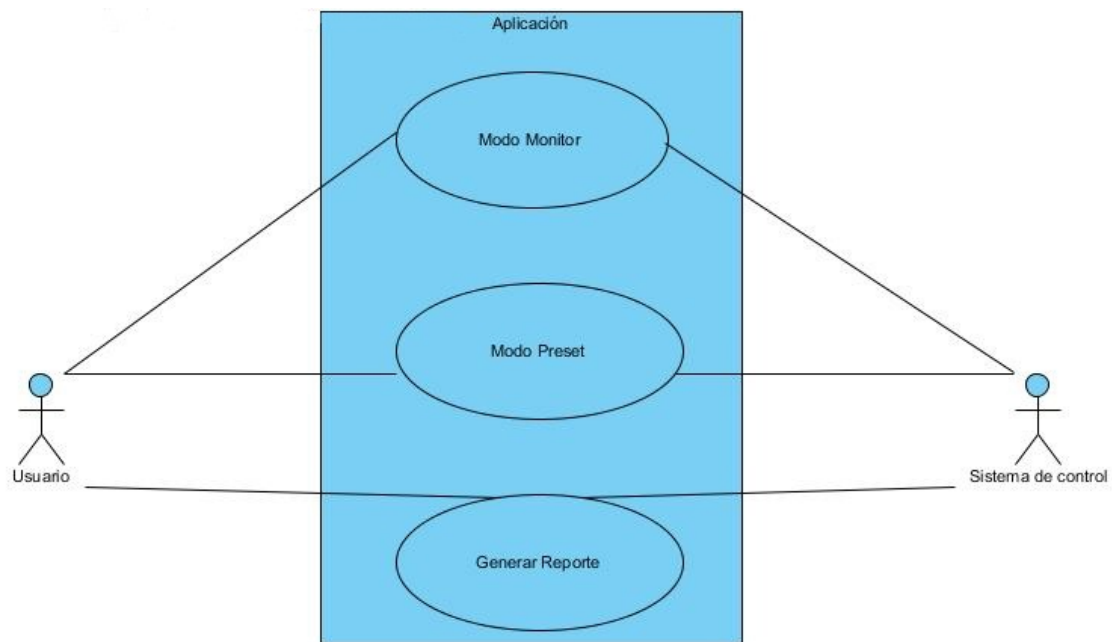
Requerimientos de usuario

A continuación se presentan los requerimientos funcionales esperados por el usuario.

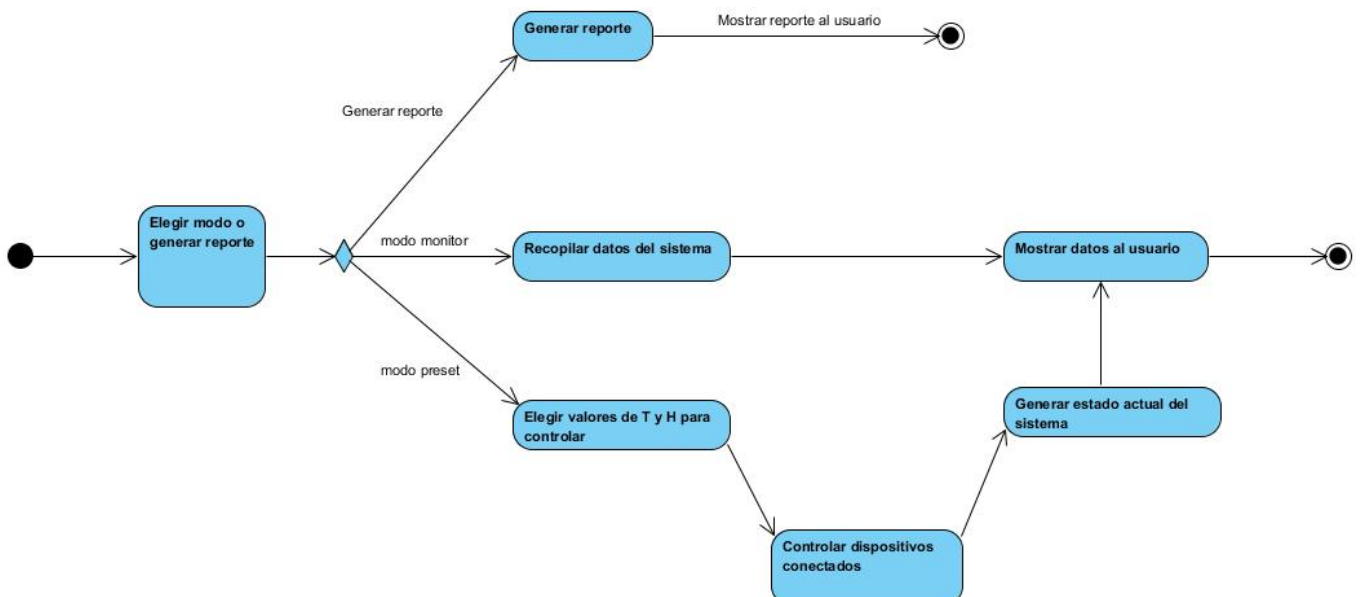
- **Modo monitor**: La aplicación le debe ofrecer al usuario un modo monitor en el cual se muestra el estado instantáneo del sistema con los valores recolectados y mostrados clara y sencillamente en la interfaz gráfica.
- **Modo preset**: El usuario debe poder setear el sistema en modo preset y con valores ingresados por el usuario, el sistema debe responder automáticamente.
 - En una caja de texto se podrá cargar un valor de temperatura deseada que se encuentre dentro de un rango válido, y este se enviara mediante un botón “enviar” al sistema para que el mismo opere encendiendo o apagando los periféricos necesarios, y alcance dicho valor.
 - En una caja de texto se podrá cargar un valor de humedad deseada y este se enviara al sistema mediante un botón “enviar”. El sistema deberá operar hasta alcanzar el valor deseado.
- **Generar reporte**: Se debe poder generar un reporte del sistema con el estado de los dispositivos conectados y las lecturas de los sensores de humedad y temperatura.

Diagramas de UML

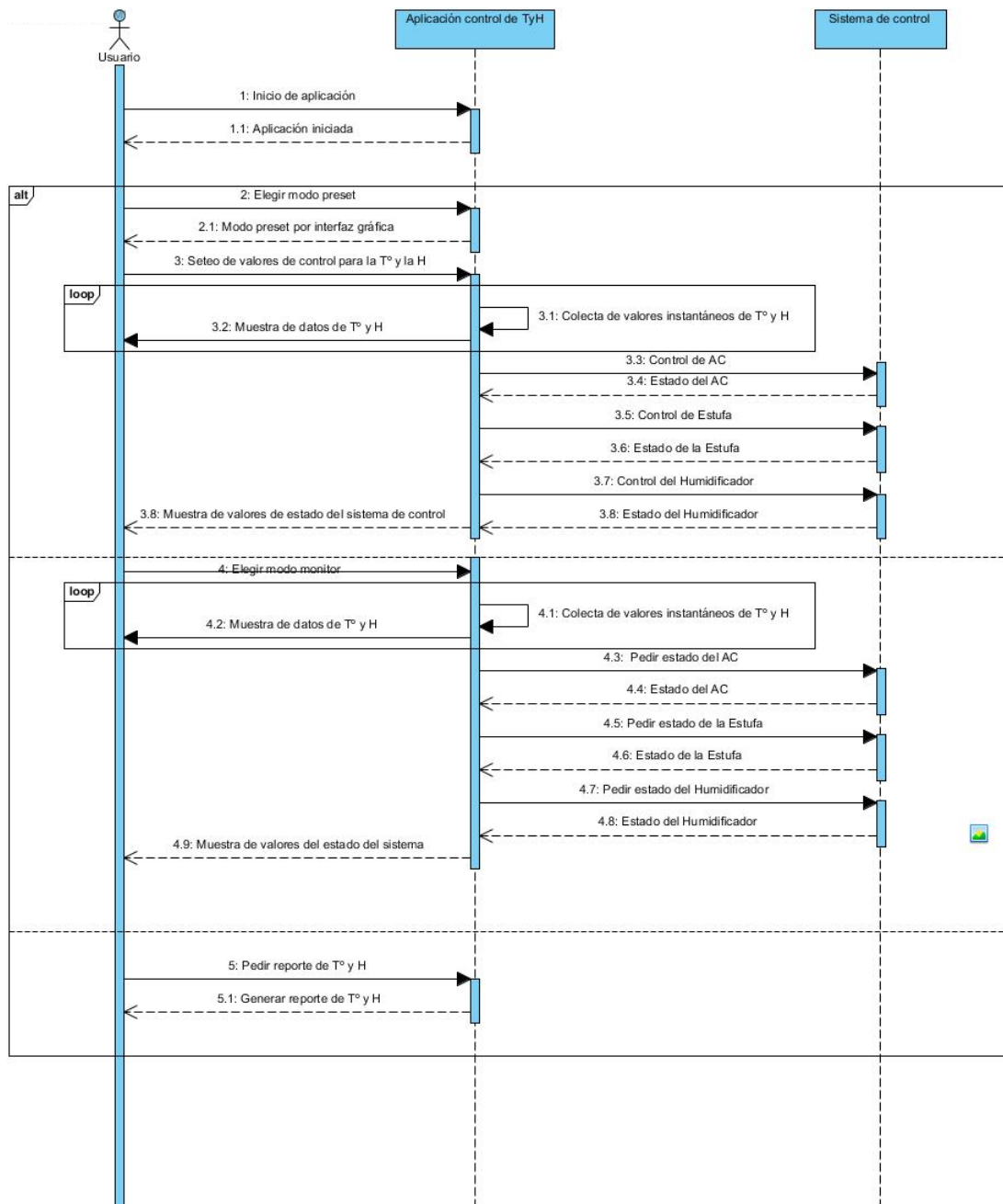
A continuación se presenta un diagrama de casos de uso para las distintas posibilidades que se le presentan al usuario.



El siguiente diagrama de actividades ilustra el flujo del programa, en el cual el usuario elige si desea generar un reporte o visualizar el modo monitor o preset y la cadena de actividades que se desenlaza.

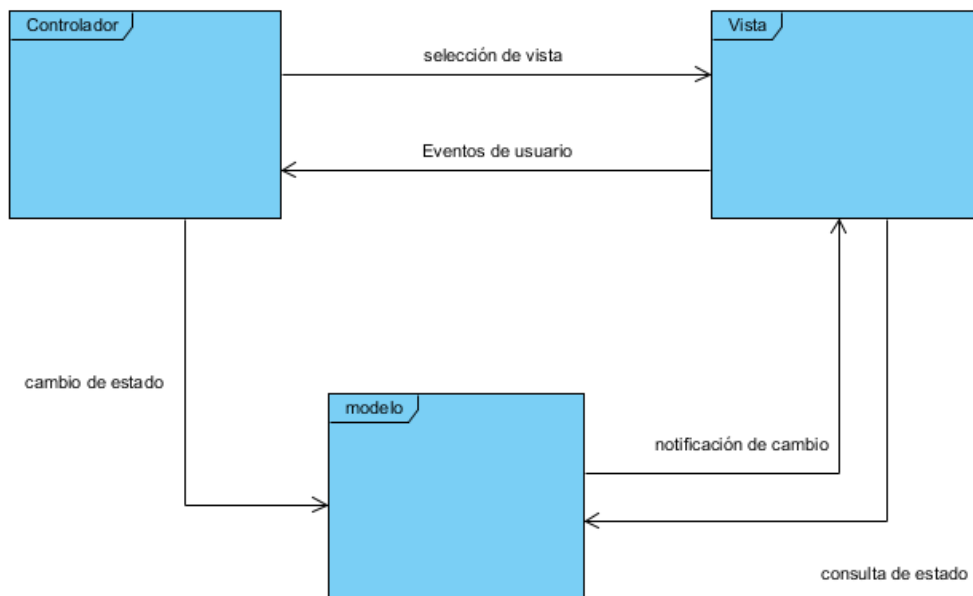


El siguiente diagrama de secuencias ilustra el sistema en los distintos casos de uso y cómo se comporta.



Arquitectura del sistema

Se implementará el MVC cuya estructura se muestra en la figura.



Especificación de los requerimientos de sistema

Requerimientos funcionales:

- **Modo monitor:** La aplicación generará una interfaz gráfica en la que se presentará, en forma de botón interactivo, el estado de cada uno de los periféricos conectados al hardware. Estos estados estarán almacenados como variables en programa en Java. El AC, la estufa eléctrica y el humidificador de ambiente. Se presentarán en tiempo real los valores recolectados por el sensor de temperatura y humedad y se los presentará en un casillero con una precisión de 1 decimal.
- **Modo preset:** Se generará una interfaz gráfica muy similar al modo monitor. Se le permite al usuario ingresar valores de temperatura y humedad deseados y el sistema correrá un bloque de código encargado de tomar decisiones para controlar el sistema de periféricos conectados al sistema hosteado en una Raspberry Pi modelo 3b. Una vez realizado el control de los dispositivos conectados, se actualizará el valor de las variables declaradas en Java que almacenan el estado de los dispositivos. Luego se muestra en la interfaz gráfica el valor de dichas variables.
- **Generar reporte:** Se genera un reporte del estado instantáneo del sistema. Esto se logra levantando el valor almacenado de las variables de estado de los periféricos en el programa en Java y relevando los datos necesarios del sensor de temperatura y humedad DHT11. Estos datos son exportados en un archivo en formato .txt.

Requerimientos no funcionales:

- El tiempo de respuesta de los dispositivos conectados al sistema no debe ser mayor a 5 segundos.
- La interfaz debe ser fácil e intuitiva y se espera que para cualquier usuario con un mínimo conocimiento de computación pueda manejarlo sin mayores dificultades luego de usar el sistema por un tiempo aproximado de 1 hora.
- La respuesta de la interfaz gráfica no debe ser mayor a 0.5 segundos.
- El sistema debe ser portable y correr en cualquier dispositivo que pueda ejecutar códigos en Java con su respectiva máquina virtual. Se desarrollará para Windows 7 en adelante para su versión de escritorio.
- El sistema debe ser eficiente y tener un desempeño ágil en hardware poco potente. Se espera que no ocupe un espacio en disco mayor a 50 mb, no utilice más de 100 mb de RAM y pueda ser corrido por un CPU de gama baja de un núcleo de procesamiento.
- El sistema deberá permitir la actualización y la incorporación de nuevas funciones de domótica a partir de la primera versión.

Lista de requerimientos

REQ1: “visualizarEstado” El sistema debe proveer una interfaz gráfica que muestre los valores de los sensores en todo momento.

REQ2: “setearTemperatura” El sistema debe permitir el ingreso de un valor de temperatura deseado y operar con los periféricos para alcanzar dicho valor

REQ3: “setearHumedad” El sistema debe permitir el ingreso de un valor de humedad deseado y operar con el periférico correspondiente para alcanzarlo.

REQ4: “generarTxt” El sistema debe tener en la interfaz gráfica un botón que permita guardar un reporte del estado actual de los sensores y periféricos en formato .txt

Matriz de trazabilidad

	PRUEBAS UNITARIAS			PRUEBAS DE SISTEMA			
	SetT	SetH	GenTxt	VIG	STD	SHD	GR
req1							
req2							
req3							
req4							

REFERENCIAS	
SetT(x) = setearTemperatura	VIG = Visualizacion de estado de dispositivos
SetH(x) = setearHumedad	STD = Seteo de temperatura deseada
GenTxt = generarTxt	SHD = Seteo de humedad deseada
	GR = Generar reporte .txt
REQ1: “visualizarEstado”	
REQ2: “setearTemperatura”	
REQ3: “setearHumedad”	
REQ4: “generarTxt”	

Diseño de casos de pruebas de sistema

Pruebas de uso normal

Pruebas para validar requerimientos funcionales

- *Prueba vs Modo monitor:* Se implementará una prueba que seleccione el modo monitor y compare los valores entregados al usuario por medio de la interfaz gráfica con los valores almacenados en el código del programa en Java. Además, validará los estados de los componentes conectados al sistema con los valores de las variables del programa
- *Prueba vs Modo preset:* Se generarán valores aleatorios de temperatura y humedad dentro de un rango aceptable y se los ingresará al programa. Luego se validará que el sistema tome las decisiones adecuadas de automatización sobre los periféricos conectados.
- *Prueba vs Generar reporte:* La prueba hará generar al programa un reporte con valores predeterminados y luego se validará que el archivo generado sea un .txt con los valores adecuados y esperados de un reporte.

Pruebas para validar requerimientos no funcionales

- *El tiempo de respuesta de los dispositivos conectados al sistema no debe ser mayor a 5 segundos:* Para verificar este requerimiento se enviará una señal a cada dispositivo y se contará que el tiempo de respuesta no supere los 5 segundos para que la prueba sea positiva.
- *La respuesta de la interfaz gráfica no debe ser mayor a 0.5 segundos:* Para testear este requerimiento se operará con los botones de la GUI y se contará que la respuesta de estos sea siempre menor a 0.5 segundos.
- *El sistema debe ser eficiente y tener un desempeño ágil en hardware poco potente. Se espera que no ocupe un espacio en disco mayor a 50 mb, no utilice más de 100 mb de RAM y pueda ser corrido por un CPU de gama baja de un núcleo de procesamiento:* Para esta prueba, la aplicación deberá correr verificando en el monitor de recursos del sistema que se cumpla con los parámetros requeridos.

Pruebas de uso extremo y de errores no esperados

- Se implementará una prueba que en modo preset, ingrese valores de humedad y temperatura fuera del rango esperable o razonable y se espera que el sistema descarte dichos valores y no realice cambios en los dispositivos conectados.
- Se implementará una prueba que en modo preset, ingrese caracteres inválidos o caracteres en blanco en los campos de humedad y temperatura y se espera que el sistema no realice cambios.

De manera preventiva, antes del diseño de las pruebas de software, se considera que las pruebas de requerimientos funcionales deben ser tomadas como Sanity Tests. Esto se encuentra sujeto a cambios a medida que se diseñen las pruebas de software.