

Requerimientos

Requirement Shell

Requirement #: 1

Requirement Type:
Funcional

Event/Use Case #:

Description: Se debe permitir capturar datos por medio de los sensores.

Rationale: El uso de sensores permite mantener un constante monitoreo de las diferentes variables necesarias para controlar un viñedo.

Source: Fabian Giraldo

Fit Criterion: Dado que el usuario entre a la aplicación debe poder ver información sobre los sensores que hay en el viñedo.

Customer Satisfaction: 5 Customer Dissatisfaction:4

Dependencies:

Supporting Materials:

History: Creado en Marzo 2, 2019.

Requirement Shell

Requirement #: 2

Requirement Type:
Funcional

Event/Use Case #:

Description: La aplicación debe enviar los datos que recoge de los sensores a una base de datos.

Rationale: Los datos recogidos por el sensor deben ser almacenados para luego poder ser mostrados al usuario.

Source: Fabian Giraldo

Fit Criterion: Al hacer una consulta en la base de datos esta debe devolver los datos solicitados referentes a los sensores.

Customer Satisfaction: 5 Customer Dissatisfaction:4

Dependencies: 1

Supporting Materials:

History: Creado en Marzo 2, 2019.

Requirement Shell

Requirement #: 3

Requirement Type:
Funcional

Event/Use Case #:

Description: La aplicación debe permitir ingresar datos, que no sean capturados por los sensores, de forma manual por el usuario.

Rationale: Existe información que no es posible capturar por medio de los sensores y que solo el usuario puede determinar, cómo por ejemplo la calidad de las uvas.

Source: Fabian Giraldo

Fit Criterion: El usuario al entrar en la aplicación debe poder acceder a un área donde exista un formulario para ingresar datos de forma manual.

Customer Satisfaction: 5 Customer Dissatisfaction:3

Dependencies:

Supporting Materials:

History: Creado en Marzo 2, 2019.

Requirement Shell

Requirement #: 4

Requirement Type:
Funcional

Event/Use Case #:

Description: La aplicación debe ser capaz de enviar alarmas al usuario basado en los datos almacenados.

Rationale: Es importante para el usuario saber cuando una variable cambia para mantener control sobre el viñedo.

Source: Fabian Giraldo

Fit Criterion: Cuando una variable pase un nivel determinado el usuario debe recibir una alarma sobre este evento.

Customer Satisfaction:4

Customer Dissatisfaction:3

Dependencies: 6

Supporting Materials:

History:

Requirement Shell

Requirement #: 5

Requirement Type:
Funcional

Event/Use Case #:

Description: La aplicación debe mostrar los datos obtenidos por medio de los sensores.

Rationale: Los datos obtenidos se tienen que mostrar para que se pueda llevar un correcto seguimiento del cultivo en el viñedo.

Source: Fabián Giraldo

Fit Criterion: Dado que el agricultor esté en la página principal y cuando seleccione una parcela en específico entonces los datos mostrados en pantalla deben ser correctos por cada sensor o los metidos manualmente.

Customer Satisfaction: 5

Customer
Dissatisfaction: 5

Dependencies: 1, 2, 3

Supporting Materials:

History: Creado en Marzo 2, 2019.

Requirement Shell

Requirement #: 6

Requirement
Type: Funcional

Event/Use Case #:

Description: La aplicación debe permitir la configuración de eventos según la necesidad del usuario.

Rationale: El manejo de eventos permite la personalización de las alarmas que van a notificarse al agricultor, debido al conocimiento en específico que tiene éste para el cuidado de su cultivo.

Source: Fabián Giraldo

Fit Criterion: Dado que el agricultor se encuentre en la página principal y cuando éste seleccione la opción de configurar eventos, entonces el programa deberá permitirle fijar el límite tanto superior como inferior de los datos recolectados en el cultivo.

Customer Satisfaction: 5

Customer
Dissatisfaction: 4.5

Dependencies:1, 2, 3

Supporting Materials:

History: Creado en Marzo 2, 2019.

Requirement Shell

Requirement #: 7 Requirement Event/Use Case #:
Type: Funcional

Description: La aplicación debe permitir crear, leer, actualizar y borrar de la base de datos a los agricultores.

Rationale: El registro y manejo de los agricultores es necesario para que únicamente los agricultores registrados puedan tener acceso a los datos obtenidos por los sensores en su respectivo cultivo.

Source: Fabián Giraldo

Fit Criterion: Dado que el agricultor se encuentre en la página principal entonces cuando éste seleccione la opción de administrar usuarios, entonces el programa deberá permitirle crear, ver, borrar o actualizar los datos de éstos.

Customer Satisfaction: 4 Customer Dissatisfaction: 4

Dependencies:

Supporting Materials:

History: Creado en Marzo 2, 2019.

Requirement Shell

Requirement #: 8 Requirement Type: Event/Use Case #:
Funcional

Description: La aplicación debe permitir crear, leer, actualizar y borrar de la base de datos a los sensores.

Rationale: El registro y manejo de los sensores es necesario para que el usuario pueda hacer una correcta administración de estos según el uso que se les esté dando.

Source: Fabián Giraldo

Fit Criterion: Dado que el usuario esté en la pantalla donde se encuentran los diferentes sensores, este debe poder ver, agregar, editar y eliminar sensores.

Customer Satisfaction: 5 Customer Dissatisfaction: 5

Dependencies:

Supporting Materials:

History: Creado en Marzo 2, 2019.

Requirement Shell

Requirement #: 9 Requirement Event/Use Case #:
Type: Funcional

Description: La aplicación debe permitir crear, leer, actualizar y borrar de la base de datos las parcelas.

Rationale: El registro y manejo de las parcelas es necesario para que el usuario pueda hacer una correcta administración de estas según el uso que se les esté dando.

Source: Fabián Giraldo

Fit Criterion: Dado que el agricultor se encuentre en la página principal entonces cuando éste seleccione la opción de administrar parcelas, entonces el programa deberá permitirle crear, ver, borrar o actualizar los datos de éstas.

Customer Satisfaction: 4 Customer Dissatisfaction: 4

Dependencies:
Supporting Materials:
History: Creado en Marzo 2, 2019.

Requirement Shell

Requirement #: 10

Requirement Type: No
funcional

Event/Use Case #:

Description: La aplicación debe mostrar cada uno de los datos mostrados en pantalla en tiempo real.

Rationale: La visualización de datos en tiempo real permite al agricultor tener conocimiento actual del estado del cultivo en cualquier momento.

Source: Fabián Giraldo

Fit Criterion: Dado que el agricultor esté en la página principal y cuando seleccione una parcela en específico entonces los datos mostrados en pantalla deben ser precisos, actuales y correctos por cada sensor o datos medidos manualmente.

Customer Satisfaction: 5

Customer

Dissatisfaction: 4

Dependencies: 1, 2, 3, 5

Supporting Materials:

History: Creado en Marzo 2, 2019.

Requirement Shell

Requirement #: 11

Requirement Type: No
funcional

Event/Use Case #:

Description: La aplicación debe mostrar gráficos de cada una de las variables tomadas por cada sensor

Rationale: El uso de gráficos le permite al usuario analizar de manera más sencilla el estado de las variables estudiadas en determinado momento.

Source: Fabian Giraldo

Fit Criterion: Dado que el agricultor esté en la página principal, cuando seleccione una parcela en específico entonces se le mostrará un gráfico con la variación de la condición climática capturada por el sensor.

Customer Satisfaction: 5

Customer

Dissatisfaction: 4

Dependencies: 1, 2, 5

Supporting Materials:

History: Creado en Marzo 2, 2019

Requirement Shell

Requirement #: 12

Requirement Type: No
funcional

Event/Use Case #:

Description: La aplicación debe permitir al usuario visualizar los datos correspondientes solo de la parcela seleccionada.

Rationale: Para el usuario es más cómodo visualizar la información perteneciente a una parcela a la vez para poder llevar a cabo una gestión de manera más sencilla.

Source: Fabian Giraldo

Fit Criterion: Dado que el agricultor esté en la página principal cuando seleccione una parcela en específico entonces se le mostrarán los gráficos generados por los datos tomados únicamente en esta parcela.

Customer Satisfaction: 4 Customer
Dissatisfaction: 3

Dependencies: 1, 2, 5, 10

Supporting Materials:

History: : Creado en Marzo 2, 2019

Requirement Shell

Requirement #: 13 Requirement Type: No funcional Event/Use Case #:

Description: La aplicación debe tener una interfaz gráfica amigable.

Rationale: El usuario prefiere una interfaz agradable e intuitiva que mejore su experiencia y a la vez facilite el uso de la plataforma.

Source: Fabian Giraldo

Fit Criterion: Dado que el agricultor esté registrado, cuando ingrese a la aplicación entonces se le ofrecerá una interfaz gráfica amigable.

Customer Satisfaction: 5 Customer
Dissatisfaction: 4

Dependencies:

Supporting Materials:

History: : Creado en Marzo 2, 2019

Requirement Shell

Requirement #: 14 Requirement Type: No funcional Event/Use Case #:

Description: Los sensores deben funcionar 24/7 por lo que la captura de datos debe darse también de esta manera.

Rationale: Es necesario responder a todos los posibles eventos dados en cualquier momento y que a su vez el usuario pueda visualizar los datos en el momento que lo desee.

Source: Fabian Giraldo

Fit Criterion: Dado que los sensores se encuentren debidamente ubicados, cuando se haya hecho la conexión con la base de datos y el frontend entonces los datos podrán ser visualizados por el agricultor 24/7.

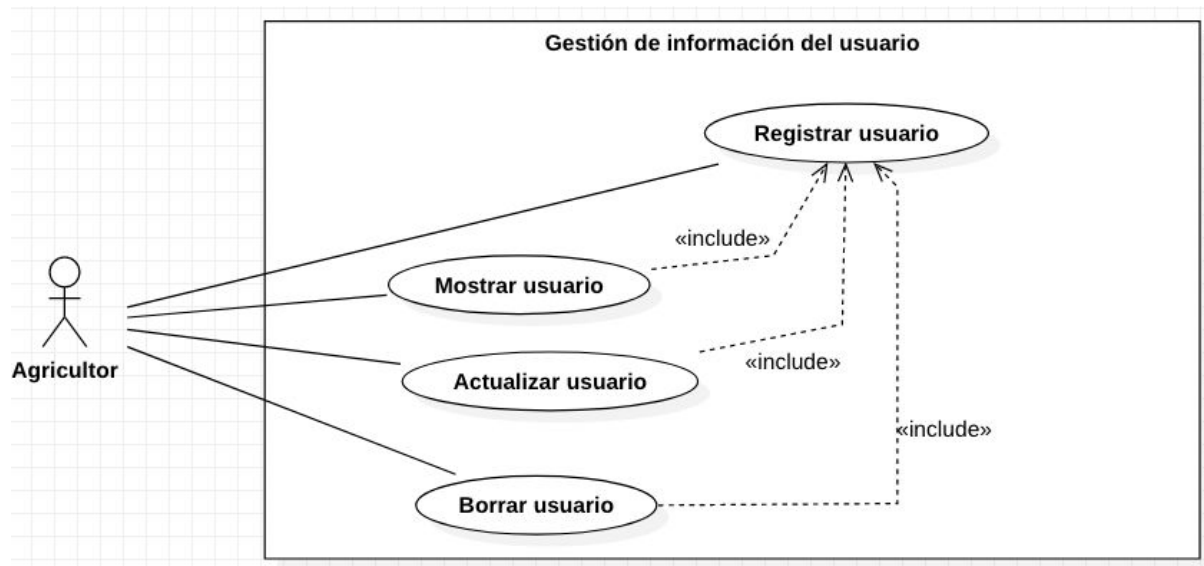
Customer Satisfaction: 5 Customer
Dissatisfaction: 5

Dependencies: 1, 2, 5, 10

Supporting Materials:

History: : Creado en Marzo 2, 2019

Casos de Uso



Use Case: Crear Agricultores

ID: 1

Brief Description: El usuario podrá registrarse en la aplicación como un nuevo agricultor.

Primary Actors: Usuario

Secondary Actors:

Preconditions: El usuario no debe tener un perfil ya registrado en el sistema.

Main Flow:

1. El caso de uso empieza cuando un nuevo usuario quiere registrarse en el sistema.
2. El usuario debe ingresar los datos solicitados por el sistema y oprimir el botón para confirmar estos datos.
3. El caso de uso termina cuando el usuario confirme su cuenta por medio de un correo electrónico que se le ha enviado.

Postconditions: El usuario obtiene acceso a la aplicación.

Alternative Flows: Ninguno.

Use Case: Mostra información de agricultor

ID: 2

Brief Description: El usuario podrá ver cada uno de sus datos personales.
Primary Actors: Usuario
Secondary Actors:
Preconditions: El Agricultor se debe encontrar registrado en la aplicación.
Main Flow: <ol style="list-style-type: none"> 1. El usuario ingresa en la aplicación. 2. El usuario selecciona la opción configurar perfil.
Postconditions: <ol style="list-style-type: none"> 1. Se lista correctamente cada uno de los datos del usuario.
Alternative Flows: <ol style="list-style-type: none"> 1. El usuario no se encuentra registrado. 2. El usuario se registra en el inicio de la aplicación. 3. El usuario selecciona la opción configurar perfil.

Use Case: Actualizar información de agricultor
ID: 3
Brief Description: El usuario podrá actualizar la información perteneciente a su perfil
Primary Actors: Usuario
Secondary Actors:
Preconditions: El Agricultor se debe encontrar registrado en la aplicación.
Main Flow: <ol style="list-style-type: none"> 1. El usuario ingresa en la aplicación. 2. El usuario selecciona la opción de configuración de perfil. 3. El usuario edita los campos que desee cambiar. 4. El usuario selecciona la opción de guardar configuración.
Postconditions: 1. La información editada por el usuario debe quedar actualizada en la base de datos.
Alternative Flows:

Use Case: Borrar Agricultores
ID: 4
Brief Description: El usuario podrá darse de baja de la aplicación.

Primary Actors: Agricultor
Secondary Actors:
Preconditions: El Agricultor debe tener un perfil ya registrado en el sistema.
Main Flow: <ol style="list-style-type: none">1. El caso de uso empieza cuando un Agricultor ingresa a su perfil y oprime el botón de eliminar cuenta.2. El caso de uso termina cuando en la ventana emergente el Agricultor confirma que desea eliminar su cuenta.
Postconditions: El usuario pierde el acceso a la aplicación.
Alternative Flows: Ninguno.