Backlog

1. Registrar mascota.
2. Modificar datos de la mascota.
3. Eliminar datos de la mascota.
4. Mostrar datos de la mascota.
5. Envía datos al dispositivo.
6. Envía notificación para actualizar datos de la mascota.
7. Envía notificación de aviso cuando haya poca cantidad de alimento.
8. Dispensar la cantidad de alimento a soltar.
9. Llenar el plato de comida de la mascota.
10. Abrir puerta del alimento.
11. Cerrar puerta del alimento.
12. Detectar insuficiencia de alimento del dispensador.
13. Diseño del prototipo.
14. Diseño de la base de datos.
15. Diagrama entidad-relación
16. Storyboards de aplicación de escritorio.

**Historias de usuario**

**HU01 – Registrar Mascota**

**Como:** Usuario

**Puedo:** Registrar los datos de la mascota al sistema.

**Debido a:** Qué para dispensar el alimento es necesario el registro de la mascota.

**Comprobación:**

Se realiza el registro de la mascota correctamente.

**ROI 10/5=2**

10

5

**HU02 – Modificar datos Mascota**

**Como:** Usuario

**Puedo:** Modificar los datos de la mascota al sistema.

**Debido a:** Si hay un error o ya es necesario el cambio de datos.

**Comprobación:**

Se realiza el cambio de los datos de la mascota correctamente.

**ROI 10/5=2**

10

5

**HU03 – Eliminar Mascota**

**Como:** Usuario

**Puedo:** Eliminar a la mascota

**Debido a:** Si ya no tienen a la mascota a su cuidado.

**Comprobación:**

Se eliminan los datos de la mascota correctamente.

**ROI 10/5=2**

10

5

**HU04 – Mostar Mascota**

**Como:** Usuario

**Puedo:** Mostrar los datos de la mascota

**Debido a:** Si necesito ver los datos o para actualizar es necesario mostrarlos.

**Comprobación:**

Se mostraran los datos de la mascota correctamente.

**ROI 10/5=2**

10

5

**HU05 – Envía datos al dispositivo.**

**Como:** Sistema

**Puedo:** Enviar los datos guardados al dispositivo de dispensar.

**Debido a:** Es necesario los datos para saber en qué tiempo y cuanto se va dispensar el alimento.

**Comprobación:**

Se envía y recibe los datos correctamente.

**ROI 10/4=2.5**

10

4

**HU06 –Envía notificación para actualizar datos.**

**Como:** Arduino

**Puedo:** Enviar notificación mediante SMS al dueño.

**Debido a:** Que en cierto tiempo la mascota crecerá y es necesario actualizar datos.

**Comprobación:**

Se envía y recibe el SMS correctamente.

**ROI 9/3=3**

9

3

**HU07 – Envía notificación de aviso cuando haya poca cantidad de alimento**

**Como:** Arduino

**Puedo:** Notificar al dueño con un mensaje de texto SMS que la cantidad de comida en el dispensador está por terminarse.

**Debido a:** Para dispensar la comida de la mascota es necesario contar con alimento suficiente.

**Comprobación:**

Se enviara un mensaje de aviso cuando quede poco alimento.

**ROI 8/3=2.6**

7

10

**HU08 – Dispensar la cantidad de alimento a soltar.**

**Como:** Arduino

**Puedo:** Soltar la cantidad de comida que la mascota necesita.

**Debido a:** El dueño configura y coloca los datos de la mascota para que el sistema la proporcione de forma automática.

**Comprobación:**

Dispensar la comida a la mascota dependiendo de las características del ejemplar.

**ROI 10/7=1.4**

3

8

**HU09 –** **Llenar el plato de comida de la mascota**

**Como:** Arduino

**Puedo:** Proporcionar la comida de la mascota en su plato o recipiente correspondiente.

**Debido a:** Que es necesario dispensar la comida de la mascota en una zona, se realiza en su plato personal.

**Comprobación:**

Llenar el plato de comida de la mascota con su comida adecuada.

**ROI 10/7=1.4**

10

7

**HU10 –** **Abrir puerta del alimento.**

**Como:** Arduino

**Puedo:** Abrir la compuesta que permite la salida de comida del contenedor al plato de la mascota.

**Debido a:** Que existirá una compuerta que bloquee el acceso de comida y solo se accione cuando se requiera proporcionar comida.

**Comprobación:**

Se abrirá la compuerta del recipiente para dispensar la comida de la mascota.

**ROI 10/7=1.4**

7

10

7

10

**HU11 –** **Cerrar puerta del alimento.**

**Como:** Arduino

**Puedo:** Cerrar la compuerta para impedir que se siga dispensando más comida al plato de la mascota.

**Debido a:** Debe proporcionar únicamente la comida suficiente que requiera el animal.

**Comprobación:**

Se cerrara la compuerta cuando el alimento dispensado sea el adecuado para el ejemplar.

**ROI 10/7=1.4**

**HU12 – Detectar insuficiencia de alimento del dispensador.**

**Como:** Arduino

**Puedo:** Enviar una alerta que notifique que el alimento está por acabarse

**Debido a:** Cada cierto tiempo la comida se va disminuyendo del dispensador.

**Comprobación:**

Se envía y recibe la notificación correctamente.

**ROI 10/5=2**

10

5

**HU13 – Diseño del prototipo.**

**Como: Diseñador**

**Puedo:** Crear el diseño del prototipo del dispensador.

**Debido a:** Para crear el dispensador se necesita un plano de cómo construirlo.

**Comprobación:**

Se muestra el diseño del dispensador.

**ROI 10/3=3.3**

10

3

**HU14 – Diseño de la base de datos.**

**Como:** Analista

**Puedo:** Crear la base de datos para el sistema.

**Debido a:** Se necesita guardar la información de la mascota.

**Comprobación:**

Se puede guardar la información.

**ROI 10/6=1.6**

10

6

**HU14 – Diseño de la base de datos.**

**Como:** Analista

**Puedo:** Crear la base de datos para el sistema.

**Debido a:** Se necesita guardar la información de la mascota.

**Comprobación:**

Se puede guardar la información.

**ROI 10/6=1.6**

10

6

**HU16 –Storyboards de la aplicación de escritorio.**

**Como:** Analista

**Puedo:** Usar los storyboards para crear la aplicación del sistema.

**Debido a:** necesitar una aplicación para controlar el dispensador.

**Comprobación:**

Se usará la aplicación para el manejo del sistema.

**ROI 10/2=5**

10

2