

FACULTAAD DETECNOLOGIAS Y NEGOCIOS

UNIVERSIDAD VERACRUZANA

Responsables: Guadalupe Rosas Hernández

Aldo Iban García García

Nombre del asesor: Martín Mancilla Gómez

Carrera: Ingeniería de software

Grado: 501

**CONEXIÓN DE UN ARDUINO PARA OBTENER DATOS DE
TEMPERATURA Y HUMEDAD**

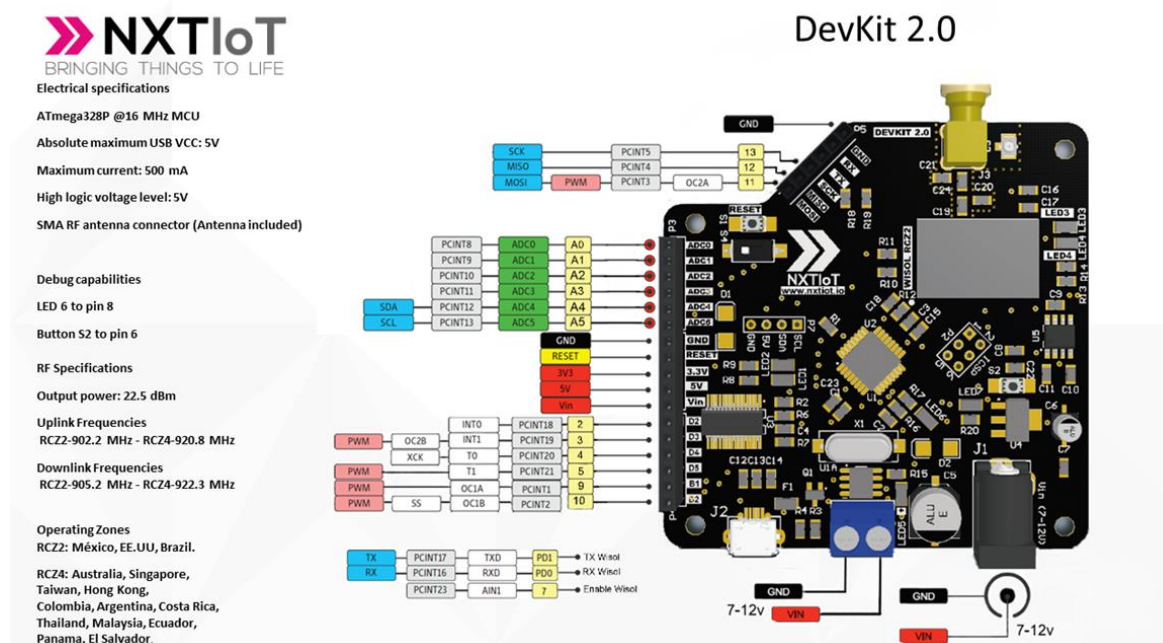
Tabla de contenido

Información general.....	2
Recursos y usuarios.....	3
Descripción detallada de los entregables.	4
Riesgos	4
Limitaciones	4
Interdependencias.	4
Aseguramiento de calidad.	4
Previsión de costos.	4
Organigrama	5
Plan de acción de trabajo.....	5
DEFINICIÓN DE ACTIVIDADES Y/O TAREAS :	5
RESPONSABLES DE LOS ENTREGABLES	6
TIEMPO DE ENTREGA.....	6
Casos de Uso	7
Diseño preliminar Mockups.....	8
Arquitectura de software.....	9
Reglas de negocio y premisas.	9
Diagrama de secuencia	10
Aspectos de usabilidad	10
°ALGUNOS LIMITES DE MENSAJE	13

Información general.

Mediante una tarjeta Arduino que es una plataforma de creación de electrónica de código abierto, la cual está basada en hardware y software libre, flexible y fácil de utilizar para los creadores y desarrolladores. Esta plataforma permite crear diferentes tipos de microordenadores de una sola placa a los que la

comunidad de creadores puede darles diferentes tipos de uso y Tiene un sistema en la nube donde desde la interfaz web es posible dar de alta los equipos, que funcionan por un ID único en vez de autenticación por tarjeta SIM como los móviles u otros dispositivos IoT basados en GPRS.



Descripción detallada de los entregables.

Se hará una conexión entre la tarjeta arduino y la página de sigfox para extraer los datos necesarios.

Riesgos

- Viento fuerte
- Falla de comunicación
- Descomposición de equipo.

Limitaciones

- Larga distancia
- Poca señal

Interdependencias.

- Arduino en caso que falle su tarjeta o esté deteriorada.
- Sigfox por si su página cae.

Aseguramiento de calidad.

Este producto está configurando con las especificaciones del usuario.

Previsión de costos.

Nombre	Costo	Inf.
Tarjeta arduino	\$250.00	Varía del tipo de tarjeta

Organigrama

Plan de acción de trabajo

	A	B	C	D	E
1	WBS	Name	Start	Finish	Duration
2	1	Extracción de datos	04/12/2019	07/01/2020	72d
3	1.1	Arduino	04/12/2019	13/12/2019	10d
4	1.1.1	Apredender arduino	04/12/2019	13/12/2019	10d
5	1.1.1.1	Contratar instructor, curso en línea o de la misma empresa	04/12/2019	06/12/2019	3d
6	1.1.1.2	Cursos de arduino	07/12/2019	13/12/2019	7d
7	1.2	Sigfox	14/12/2019	16/12/2019	10d
8	1.2.1	Aprender sigfox	14/12/2019	23/12/2019	10d
9	1.2.1.1	Contratar instructor, curso en línea o de la misma empresa	14/12/2019	16/12/2019	3d
10	1.2.1.2	Cursos de sigfox	17/12/2019	23/12/2019	7d
11	1.3	Antena	24/12/2019	16/12/2019	24d
12	1.3.1	Desarrollo de antenas	24/12/2019	16/12/2019	24d
13	1.3.1.1	Estructura de la antena	24/12/2019	29/01/2020	6d
14	1.3.1.2	Conexiones	30/12/2019	06/01/2020	8d
15	1.3.1.3	Software	07/01/2020	16/01/2020	10d
16	1.3.2	Identificar	17/01/2020	28/01/2020	12d
17	1.3.2.1	Encontrar la clave	17/01/2020	22/01/2020	6d
18	1.3.2.2	Verificar la clave	23/01/2020	28/01/2020	6d
19	1.4	Extracción	29/01/2020	07/02/2020	10d
20	1.4.1	Conectar a antena	29/01/2020	07/02/2020	10d
21	1.4.1.1	Establecer una conexión entre Arduino y la antena	29/01/2020	31/01/2020	3d
22	1.4.1.2	Hacer un desarrollo de extracción	01/02/2020	04/02/2020	4d
23	1.4.1.3	Guardar los datos de extracción	05/02/2020	07/02/2020	3d

DEFINICIÓN DE ACTIVIDADES Y/O TAREAS :

en este punto nos encargaremos de todas las definiciones de las tareas que conlleva éste proyecto

°Identificación de la temperatura ambiental

°Identificación de la humedad ambiental

°Recopilar datos captados por el sensor que son subidos al backend de SigFox

°A Través de una página agregarlos de manera gráfica toda esa información

RESPONSABLES DE LOS ENTREGABLES

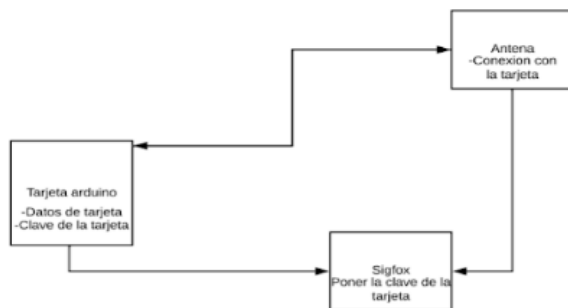
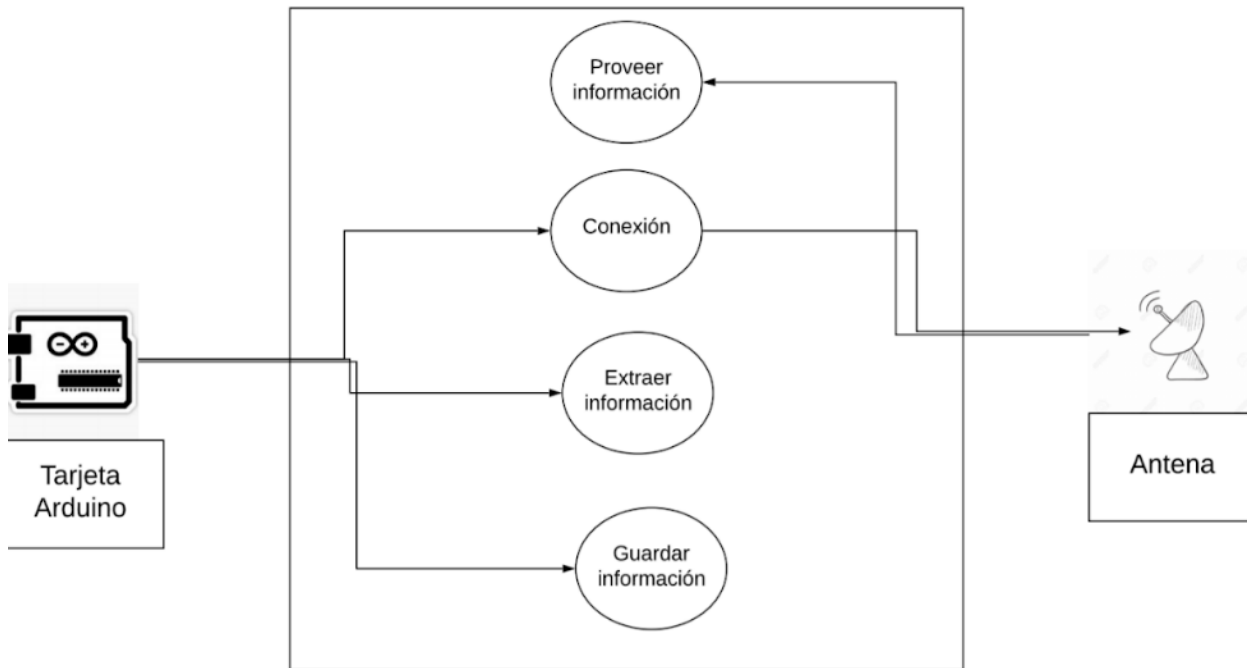
--Guadalupe Rosas Hernandez

--Aldo Ivan Garcia Garcia

TIEMPO DE ENTREGA

WBS	Name	Start	Finish	Duration	optimista	Mas probable	Pesimista	Desviacion Estandar	varianza	
1	Extracción de datos	04/12/2019	07/01/2020	107						
1.1	Arduino	04/12/2019	13/12/2019	16.33						
1.1.1	Apredender arduino	04/12/2019	13/12/2019	8.17						
1.1.1.1	Contratar instructor, curso en línea o de la misma empresa	04/12/2019	06/12/2019	2.17	1	2	4	0.50	0.25	
1.1.1.2	Cursos de arduino	07/12/2019	13/12/2019	6	4	6	8	0.67	0.44	D.E total 3.13
1.2	Sigfox	14/12/2019	16/12/2019	18.333						Tiempo Maximo 113.25
1.2.1	Aprender sigfox	14/12/2019	23/12/2019	9.1667						tiempo Minimo 100.75
1.2.1.1	Contratar instructor, curso en línea o de la misma empresa	14/12/2019	16/12/2019	2.1667	1	2	4	0.50	0.25	
1.2.1.2	Cursos de sigfox	17/12/2019	23/12/2019	7	5	7	9	0.67	0.44	
1.3	Antena	24/12/2019	16/12/2019	32.333						
1.3.1	Desarrollo de antenas	24/12/2019	16/12/2019	12.167						
1.3.1.1	Estructura de la antena	24/12/2019	29/01/2020	4.8333	3	5	6	0.50	0.25	
1.3.1.2	Conexiones	30/12/2019	06/01/2020	2	1	2	3	0.33	0.11	
1.3.1.3	Software	07/01/2020	16/01/2020	5.3333	3	5	9	1.00	1.00	
1.3.2	Identificar	17/01/2020	28/01/2020	8						
1.3.2.1	Encontrar la clave	17/01/2020	22/01/2020	4	3	4	5	0.33	0.11	
1.3.2.2	Verificar la clave	23/01/2020	28/01/2020	4	3	4	5	0.33	0.11	
1.4	Extracción	29/01/2020	07/02/2020	65						
1.4.1	Conectar a antena	29/01/2020	07/02/2020	32.5						
1.4.1.1	Establecer una conexión entre Arduino y la antena	29/01/2020	31/01/2020	10.333	7	10	15	1.33	1.78	
1.4.1.2	Hacer un desarrollo de extracción	01/02/2020	04/02/2020	12.167	8	12	17	1.50	2.25	
1.4.1.3	Guardar los datos de extracción	05/02/2020	07/02/2020	10	5	10	15	1.67	2.78	

Casos de Uso



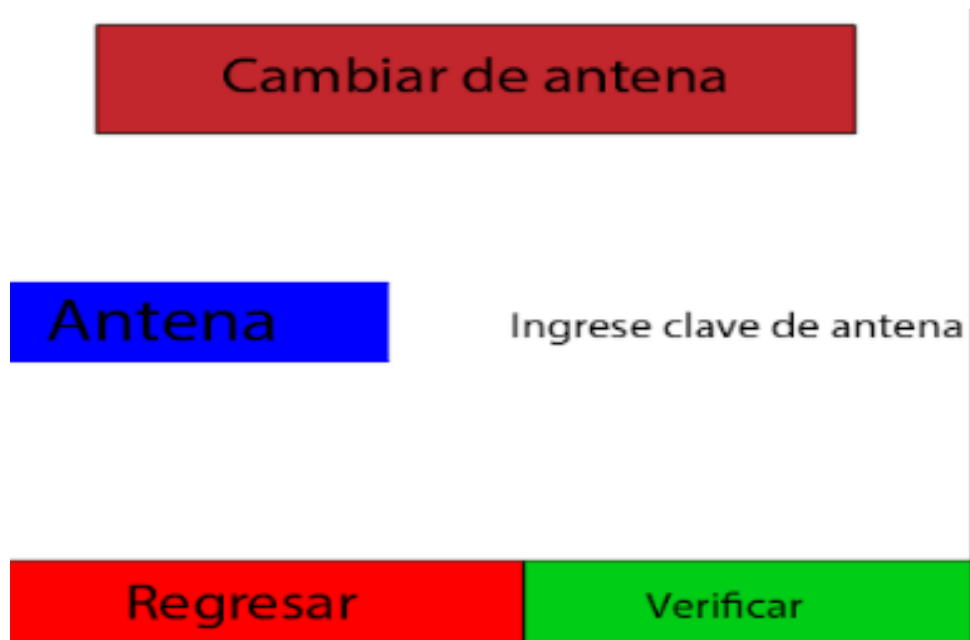
Diseño preliminar Mockups.



Acceder a extracción
de datos

Cambiar de antena

Salir



Cambiar de antena

Antena

Ingrese clave de antena

Regresar

Verificar

Extracción de datos

Temperatura	-----	5
Humedad	-----	5
Calor existente	-----	5
	-----	5
	-----	5
	-----	5

Regresar

Arquitectura de software.

Usamos c++ porque toda su estructura y ejemplos se parece a c++, entonces es más fácil, y cómodo poder desarrollar un software para extracción de datos con arduino usando la página de sigfox para este programa.

Reglas de negocio y premisas.

- Tener cuidado con los datos que se ponen en sigfox
- Verificar que la clave sea correcta

-

Diagrama de secuencia

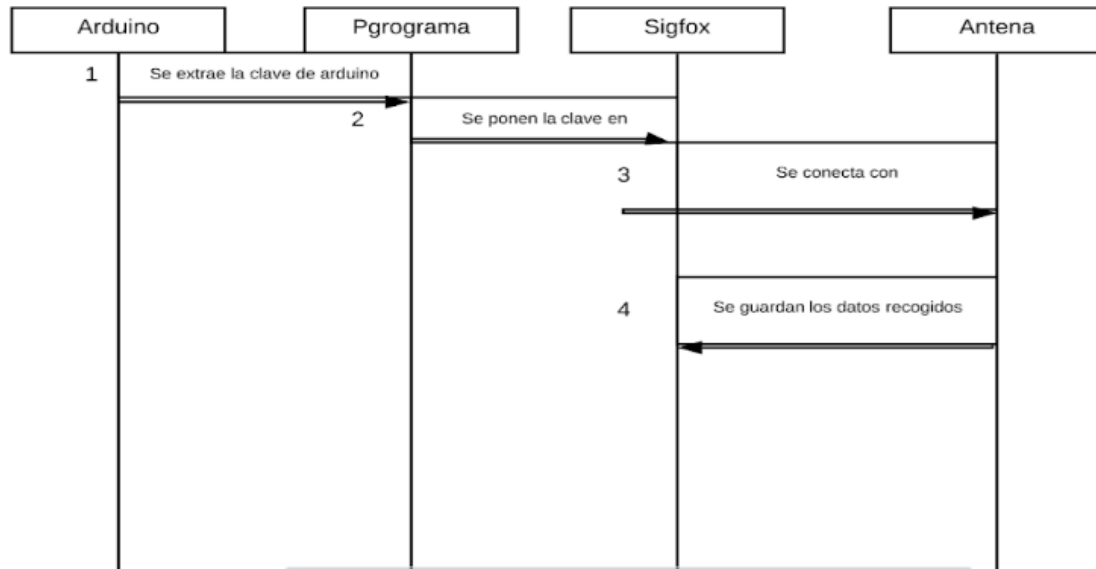


Diagrama de clases

Aspectos de usabilidad

Es fácil de usar porque la página de sigfox te lo facilita todo, entonces solo tienes que usar el programa para extraer la clave de la tarjeta arduino para saber a qué tarjeta se conecta y después hacer una conexión hacia la antena.

para obtener el id y el pac necesitamos un programa llamado “devkit_info.ino” y así poderlo dar de alta, nos registramos en la página de sigfox ingresamos las credenciales que credenciales

Device - List

Count: 1 / 1

page 1

Communication status	Device type	Group	Id	Last seen	Name	Token state
	DT_413E25	Martin Mancilla	413E25	2019-10-08 14:37:34	Devkit_413E25	

page 1

Copyright © Sigfox - 8.10.0-63c7726-20191007.085239 - 290 - Terms and conditions / Cookie policy.

pudiendo observar los mensajes dados

Device 413E25 - Messages

page 1

Time	Data / Decoding	LQI	Callbacks	Location
2019-10-08 14:37:34	0102030405 Mensaje: 1			
2019-10-08 14:37:25	0102030405 Mensaje: 1			
2019-10-08 14:30:14	0102030405 Mensaje: 1			
2019-10-08 14:26:44	0102030405 Mensaje: 1			
2019-10-08 14:22:38	0102030405 Mensaje: 1			
2019-10-08 14:20:08	0102030405 Mensaje: 1			

Copyright © Sigfox - 8.10.0-63c7726-20191007.085239 - 290 - Terms and conditions / Cookie policy.

teniendo como código para el mensaje, usando la biblioteca proporcionada por NxTIoT.

27 lines (20 sloc) | 517 Bytes

```
1  /* Envio de mensaje a sigfox usando las librerias de NxtIoT*/
2
3  #include <NXTIoT_dev.h>
4
5  NXTIoT_dev  mysigfox;
6  const int boton=6;
7  const int led6=5;
8
9  void setup() {
10     pinMode(boton, INPUT);
11     pinMode(led6, OUTPUT);
12     Serial.begin(9600);
13 }
14
15 void loop() {
16     /*El mensaje se envia al precionar el botón azul de la placa*/
17     if (digitalRead(boton)==LOW)
18     {
19
20         mysigfox.initpayload();
21         mysigfox.addint(117);
22         mysigfox.sendmessage();
23
24         Serial.println("Mensaje enviado");
25     }
26 }
```

°ALGUNOS LIMITES DE MENSAJE

27 lines (20 sloc) | 517 Bytes

```
1  /* Envio de mensaje a sigfox usando las librerias de NxtIoT*/
2
3  #include <NXTIoT_dev.h>
4
5  NXTIoT_dev mysigfox;
6  const int boton=6;
7  const int led6=5;
8
9  void setup() {
10     pinMode(boton, INPUT);
11     pinMode(led6, OUTPUT);
12     Serial.begin(9600);
13 }
14
15 void loop() {
16     /*El mensaje se envia al precionar el botón azul de la placa*/
17     if (digitalRead(boton)==LOW)
18     {
19
20         mysigfox.initpayload();
21         mysigfox.addint(117);
22         mysigfox.sendmessage();
23
24         Serial.println("Mensaje enviado");
25     }
26 }
```

Se imagino que existía un limite de 140 mensajes al día, percatando que no hay limites ya que enviamos alrededor de 150 y aun podíamos seguir enviando.

Tras buscar en el foro de preguntas de sigfox encontramos que no existe un limite de mensajes por día, sin embargo, es recomendable mantenernos a un margen de 6 mensajes por hora.

asi que se comprobo si existia un limite de mensajes

28 lines (23 sloc) | 531 Bytes

```
1  /* Envio de mensaje a sigfox usando las librerias de NxtIoT*/
2
3  #include <NXTIoT_dev.h>
4
5  NXTIoT_dev mysigfox;
6  const int boton=6;
7  const int led6=5;
8
9  void setup() {
10     pinMode(boton, INPUT);
11     pinMode(led6, OUTPUT);
12     Serial.begin(9600);
13 }
14
15 void loop() {
16     /*Probamos si el limite de mensajes es 128*/
17     int x=0;
18     while(x<=127)
19     {
20         mysigfox.initpayload();
21         mysigfox.addint(x);
22         mysigfox.sendmessage();
23         x = x + 1;
24         Serial.println("Mensaje enviado: ",x);
25     }
26     /*Se enviaron aproximadamente 150 mensajes y no hubo error*/
27 }
```

haciendo la Configuración de Thinger.io obtendremos la conexion para que nos llegue los resultados de las temperaturas, como lo es en el siguiente ejemplo de

temperatura

