



**UNIVERSIDAD DE JAÉN**  
*Escuela Politécnica Superior (Jaén)*

Trabajo Fin de Máster

# **SISTEMA WEB DE MONITORIZACIÓN BASADO EN IoT PARA POTENCIAR COMPORTAMIENTOS SALUDABLES**

**Alumno/a: Cruz Rodríguez, Antonio Jesús**

Tutor/a: Dra. D<sup>a</sup>. Macarena Espinilla Estévez  
Dr. D. Javier Medina Quero

Dpto.: Informática

**Abril, 2018**





Universidad de Jaén  
Escuela Politecnica Superior de Jaén  
Departamento de informática

*Dra. D<sup>a</sup>. Macarena Espinilla Estévez*  
y *Dr. D<sup>o</sup>. Javier Mediana Quero*, tutores  
del Trabajo Fin de Máster titulado:

**SISTEMA WEB DE MONITORIZACIÓN BASADO EN IOT PARA POTENCIAR  
COMPORTAMIENTOS SALUDABLES,**

Que presenta *D. Antonio Jesús Cruz Rodríguez*, autorizan

Su presentación para defensa y evaluación en la  
Escuela Politécnica Superior de Jaén.

Jaén, abril de 2018.

El alumno:

*Antonio Jesús Cruz Rodríguez*

Antonio Jesús Cruz Rodríguez

Los tutores:

Macarena Espinilla Estévez

Javier Medina Quero



## Agradecimientos

*Me gustaría poder expresar mi más sincero agradecimiento a todas aquellas personas que de alguna u otro forma han contribuido a que haya podido realizar este trabajo fin de master.*

*En primer lugar, a mi familia, mis padres, hermanos y abuelos que siempre están ahí para apoyarme en los objetivos que me propongo.*

*En segundo lugar, a mi novia Nazaret, por el apoyo y el ánimo que cada día me aporta para que sea capaz de conseguir todo lo que me proponga.*

*En tercer lugar, a mis tutores, Macarena Espinilla Estévez y Javier Medina Quero, por el tiempo, dedicación, apoyo y confianza que han depositado en mí.*

*En cuarto lugar, a mi compañero Daniel Zafra Romero por ofrecerme una base inicial para el comienzo de este proyecto.*

*En quinto lugar, a la clínica Neurobase de Jaén por introducirme en la importancia de la temática de la que trata este trabajo.*

*Y por último, a todos los profesores que han colaborado en la enseñanza de este máster en ingeniería informática ya que su trabajo me ha permitido crecer como profesional y enriquecer mis conocimientos.*

*Muchas gracias a todos.*



# ÍNDICE DE CONTENIDO

<b>Capítulo 1</b> .....	18
1. Introducción .....	18
1.1. Motivación .....	18
1.2. Propuesta .....	19
1.3. Objetivos .....	20
1.4. Planificación temporal .....	20
1.4.1. Estimación de tiempos .....	21
1.4.2. Diagrama de Gantt .....	22
1.5. Presupuesto .....	24
1.6. Estructura de la memoria .....	25
<b>Capítulo 2</b> .....	27
2. Introducción .....	27
2.1. Arquitectura del sistema .....	27
2.2. Descripción del dispositivo IoT .....	28
2.2.1. Placa .....	29
2.2.2. Sensores .....	30
2.2.3. Alimentación .....	31
2.2.4. Ensamblado .....	31
2.3. Descripción de los datos recogidos .....	33
2.4. Soluciones propuestas .....	41
2.4.1. Para la tarea de comer .....	42
2.4.2. Para la tarea de beber .....	42
<b>Capítulo 3</b> .....	45
3. Proceso de ingeniería del software .....	45
3.1. Fases de la ingeniería del software .....	45
3.2. Especificación de requerimientos .....	45

3.2.1.	Requerimientos de la aplicación web.....	46
3.2.1.1.	Requerimientos funcionales.....	46
3.2.1.2.	Requerimientos no funcionales.....	47
3.2.2.	Requerimientos de la aplicación móvil .....	49
3.2.2.1.	Requerimientos funcionales.....	49
3.2.2.2.	Requerimientos no funcionales.....	50
3.2.3.	Requerimientos del sistema de dispositivos IoT .....	51
3.2.3.1.	Requerimientos funcionales.....	51
3.2.3.2.	Requerimientos no funcionales.....	52
3.3.	Análisis del sistema .....	52
3.3.1.	Casos de uso .....	53
3.3.1.1.	Casos de uso del sistema web .....	54
3.3.1.2.	Casos de uso de la aplicación móvil .....	64
3.3.1.3.	Casos de uso del sistema de dispositivos IoT .....	70
3.3.2.	Escenarios .....	72
3.3.2.1.	Escenarios del sistema web .....	72
3.3.2.2.	Escenarios de la aplicación móvil .....	78
3.3.2.3.	Escenarios del sistema de dispositivos IoT .....	81
3.4.	Diseño del sistema.....	82
3.4.1.	Diseño de clases.....	82
3.4.2.	Diseño de datos .....	88
3.4.3.	Diseño de la interfaz .....	98
3.4.3.1.	Metáforas.....	99
3.4.3.2.	Estilo.....	101
3.4.3.3.	Mensajes .....	104
3.4.3.4.	Storyboard de la aplicación web .....	104
3.4.3.5.	Storyboard de la aplicación móvil .....	108



D.2. Control de la actividad de comer .....	151
D.2.1. Control de la velocidad de comer .....	151
D.2.2. Consultar histórico .....	152
D.3. Control de la actividad de beber .....	154
D.3.1. Consultar estado de la tarea .....	154
D.3.2. Consultar histórico .....	154
D.4. Cierre de sesión .....	155
E. Manual de usuario de los sensores .....	157
E.1. Uso del sensor de beber .....	157
E.2. Uso del sensor de comer .....	157

## ÍNDICE DE FIGURAS

Figura 1.1.- Planificación temporal .....	21
Figura 1.2.- Diagrama de Gantt.....	23
Figura 2.1.-Arquitectura del sistema.....	28
Figura 2.2.- Placa Wemos Lolin 32 .....	30
Figura 2.3.-Sensor MPU-6050.....	30
Figura 2.4.- Batería .....	31
Figura 2.5.- Esquema de conexión de placa y sensor.....	32
Figura 2.6.-Conector JST .....	33
Figura 2.7.- Sensor ensamblado .....	33
Figura 2.8.- Rangos del sensor MPU-6050 .....	34
Figura 2.9.- Movimiento de beber agua.....	36
Figura 2.10.- Resultados obtenidos para la actividad de comer.....	42
Figura 2.11.- Resultados obtenidos para la actividad de beber.....	43
Figura 3.1.- Diagrama de casos de uso .....	54
Figura 3.2.- Diagrama frontera del sistema web.....	55
Figura 3.3.- Caso de uso: Identificación del administrador.....	56
Figura 3.4.-Caso de uso: Registro de un nuevo administrador .....	57
Figura 3.5.- Caso de uso: Recuperar contraseña de administrador .....	58
Figura 3.6.- Caso de uso: Modificación de los datos personales del administrador (seguridad) .....	59
Figura 3.7.-Caso de uso: Registrar un nuevo usuario (monitorizado) .....	60
Figura 3.8.-Caso de uso: Editar un usuario (monitorizado) .....	61
Figura 3.9.-Caso de uso: Control de tareas.....	63
Figura 3.10.-Caso de uso: Registrar un nuevo sensor .....	64
Figura 3.11.-Caso de uso: Diagrama frontera de la aplicación móvil .....	65
Figura 3.12.-Caso de uso: Identificar usuario.....	66
Figura 3.13.-Caso de uso: Cierre de sesión.....	67
Figura 3.14.-Caso de uso: Control de la actividad de comer.....	68
Figura 3.15.-Caso de uso: Control de la actividad de beber .....	69
Figura 3.16.-Caso de uso: Recibir notificación .....	70
Figura 3.17.-Diagrama frontera del sistema de dispositivos IoT.....	71
Figura 3.18.-Caso de uso: Recogida de datos .....	72

Figura 3.19.- Patrón Modelo-Vista-Controlador.....	83
Figura 3.20.- Diseño de clases del sistema web: Controlador Login.....	84
Figura 3.21.-Diseño de clases del sistema web: Controlador Admin .....	85
Figura 3.22.- Diseño de clases del sistema web: Controlador Services.....	86
Figura 3.23.- Diseño de clases de la aplicación móvil.....	87
Figura 3.24.- Diseño de clases del sistema de dispositivos IoT .....	88
Figura 3.25.- Diagramas Entidad-Relación .....	91
Figura 3.26.- Esquema conceptual de la base de datos del sistema web.....	92
Figura 3.27.- Esquema conceptual de la aplicación móvil.....	98
Figura 3.28.- Logo del sistema web .....	99
Figura 3.29.- Icono de la tarea de beber .....	99
Figura 3.30.- Icono de la tarea de comer .....	100
Figura 3.31.- Icono de dispositivos IoT.....	100
Figura 3.32.- Logo de la aplicación móvil .....	100
Figura 3.33.- Icono de perfil de usuario.....	101
Figura 3.34.- Icono de información.....	101
Figura 3.35.- Iconos de valoración de la actividad de comer .....	101
Figura 3.36.- Paleta de colores .....	102
Figura 3.37.- Iconos de Bootstrap .....	103
Figura 3.38.- Iconos de Google Desing.....	103
Figura 3.39.- Botones de Bootstrap.....	103
Figura 3.40.- Mensajes Bootstrap .....	104
Figura 3.41.-Storyboard sistema web: Inicio sesión 1 .....	105
Figura 3.42.-Storyboard sistema web: Inicio sesión 2 .....	105
Figura 3.43.-Storyboard sistema web: Estado de la tarea de beber.....	106
Figura 3.44.-Storyboard sistema web: Estado de la tarea de comer.....	106
Figura 3.45.-Storyboard sistema web: Usuarios 1.....	106
Figura 3.46.-Storyboard sistema web: Usuarios 2.....	107
Figura 3.47.-Storyboard sistema web: Usuarios 3.....	107
Figura 3.48.-Storyboard sistema web: Dispositivos IoT .....	107
Figura 3.49.-Storyboard sistema web: Administradores.....	108
Figura 3.50.-Storyboard de la aplicación móvil: Identificar usuario .....	108
Figura 3.51.-Storyboard de la aplicación móvil: Tarea de comer .....	109
Figura 3.52.-Storyboard de la aplicación móvil: Tarea de beber .....	109

Figura 3.53.-Storyboard de la aplicación móvil: Cierre de sesión.....	110
Figura 3.54.- Logo CodeIgniter.....	112
Figura 3.55.- Atom.....	113
Figura 3.56.- phpMyAdmin .....	113
Figura 3.57.- Android Studio.....	114
Figura 3.58.- Arduino IDE.....	115
Figura 3.59.- SmartGit.....	116

## ÍNDICE DE FIGURAS DE MANUALES

Figura manuales 1.- Creación de la base de datos en phpMyAdmin .....	137
Figura manuales 2.-Agregar usuario a la base de datos .....	138
Figura manuales 3.- Importar fichero .sql.....	138
Figura manuales 4.- Modificación de Wifi.....	140
Figura manuales 5.-Modificación del identificador del sensor .....	141
Figura manuales 6.- Subir código Arduino IDE .....	141
Figura manuales 7.- Formulario de acceso al sistema web .....	142
Figura manuales 8.- Panel de administración del sistema web.....	142
Figura manuales 9.- Formulario para recordar credenciales.....	143
Figura manuales 10.- Formulario de registro de un nuevo administrador .....	144
Figura manuales 11.- Registro de nuevo usuario.....	145
Figura manuales 12.- Formulario de registro de un nuevo usuario .....	145
Figura manuales 13.- Configuración de objetivos .....	146
Figura manuales 14.- Configuración de avisos .....	147
Figura manuales 15.- Información sobre las tareas de beber .....	148
Figura manuales 16.- Información sobre las tareas de comer .....	148
Figura manuales 17.- Pantalla sensores .....	149
Figura manuales 18.- Formulario de registro de un nuevo sensor .....	149
Figura manuales 19.- Identificación en la aplicación móvil.....	150
Figura manuales 20.- Pantalla principal de la aplicación móvil .....	150
Figura manuales 21.- Monitorización de las tareas de comer .....	151
Figura manuales 22.- Información sobre la velocidad de comer .....	152
Figura manuales 23.- Histórico de las tareas de comer .....	153
Figura manuales 24.-Detalle de las tareas de comer .....	153

Figura manuales 25.- Consultar estado de la tarea de beber .....	154
Figura manuales 26.- Histórico de las tareas de comer .....	155
Figura manuales 27. Perfil de usuario en la aplicación móvil .....	156
Figura manuales 28.- Confirmación de cierre de sesión en aplicación móvil .	156
Figura manuales 29.- Uso del sensor de beber.....	157
Figura manuales 30.- Uso del sensor de comer.....	158

## ÍNDICE DE TABLAS

Tabla 1.1. Planificación temporal .....	22
Tabla 1.2.- Presupuesto .....	25
Tabla 2.1.- Esquema de conexión de placa y sensor .....	32
Tabla 2.2.- Datos obtenidos para determinar la velocidad de comer .....	39
Tabla 2.3.- Casos de 20ml .....	40
Tabla 2.4.- Casos de 45 ml .....	41
Tabla 2.5.- Casos de 75ml .....	41
Tabla 2.6.- Resultados obtenidos para la actividad de beber .....	43
Tabla 3.1.- Servicios REST implementados .....	118
Tabla 3.2.- Validación del sistema web. Caso 1: Login correcto de administrador.....	119
Tabla 3.3.- Validación del sistema web. Caso 2: Login incorrecto de administrador.....	119
Tabla 3.4.- Validación del sistema web. Caso 3: Crear un nuevo administrador de forma correcta .....	120
Tabla 3.5.- Validación del sistema web. Caso 4: crear un nuevo administrador de forma incorrecta .....	120
Tabla 3.6.- Validación del sistema web. Caso 5: Registro correcto de un nuevo usuario .....	120
Tabla 3.7.- Validación del sistema web. Caso 6: Registro incorrecto de un nuevo usuario.....	120
Tabla 3.8.- Validación del sistema web. Caso 7: Editar un usuario de forma correcta .....	121
Tabla 3.9.- Validación del sistema web. Caso 8: Editar un usuario de forma incorrecta.....	121
Tabla 3.10.- Validación del sistema web. Caso 9: Control de la actividad de comer. Consultar datos (correcto).....	121
Tabla 3.11.- Validación del sistema web. Caso 10: Control de la actividad de comer. Consultar datos (incorrecto) .....	122
Tabla 3.12.- Validación del sistema web. Caso 11: Control de la actividad de beber. Consultar datos (correcto).....	122

Tabla 3.13.- Validación del sistema web. Caso 12: Control de la actividad de beber. Consultar datos (incorrecto) .....	122
Tabla 3.14.- Validación del sistema web. Caso 13: Generar de forma correcta identificador de usuario para la aplicación móvil .....	123
Tabla 3.15.- Validación del sistema web. Caso 14: Generar de forma incorrecta identificador de usuario par la aplicación móvil. Repetición de identificador en el sistema .....	123
Tabla 3.16.- Validación del sistema web. Caso 15: Avisos. Envió de aviso (correcto).....	123
Tabla 3.17.- Validación del sistema web. Caso 16: Avisos. Envió de aviso (incorrecto). Fallo de comunicación.....	124
Tabla 3.18.- Validación del sistema web. Caso 17: Registro de sensor de forma correcta .....	124
Tabla 3.19.- Validación del sistema web. Caso 18: Registro de sensor de forma incorrecta. Identificador de usuario no correcto.....	124
Tabla 3.20.- Validación del sistema web. Caso 19: Modificación de datos personales del administrador de forma correcta. ....	124
Tabla 3.21.- Validación del sistema web. Caso 20: Modificación de datos personales del administrador de forma incorrecta.....	125
Tabla 3.22.- Validación de la aplicación móvil. Caso 1: Login correcto del usuario .....	125
Tabla 3.23.- Validación de la aplicación móvil. Caso 2: Login incorrecto del usuario .....	125
Tabla 3.24.- Validación de la aplicación móvil. Caso 3: Control de la actividad de comer (correcto). Consulta de histórico.....	125
Tabla 3.25.- Validación de la aplicación móvil. Caso 4: Control de la actividad de comer (incorrecto). Consulta de histórico (Fallo en la conexión de Internet) .....	126
Tabla 3.26.- Validación de la aplicación móvil. Caso 5: Control de la actividad de comer (correcto). Monitorización de la actividad de comer .....	126
Tabla 3.27.- Validación de la aplicación móvil. Caso 6: Control de la actividad de comer (incorrecto). Monitorización de la actividad de comer (Fallo en la conexión a Internet).....	126

Tabla 3.28.- Validación de la aplicación móvil. Caso 7: Control de la actividad de beber (correcto). Consulta de historico .....	127
Tabla 3.29.- Validación de la aplicación móvil. Caso 8: Control de la actividad de beber (incorrecto). Consulta de histórico (Fallo en la conexión a Internet)	127
Tabla 3.30.- Validación de la aplicación móvil. Caso 9: Recibir notificaciones	127
Tabla 3.31.- Validación de la aplicación móvil. Caso 10: Cierre de sesión ....	128
Tabla 3.32.- Validación del sistema de sensores. Caso 1: Recogida de datos (correcto).....	128
Tabla 3.33.- Validación del sistema de sensores. Caso 2: Recogida de datos (incorrecto) .....	128
Tabla 3.34.- Validación del sistema de sensores. Caso 3: Procesamiento de datos (correcto) .....	128
Tabla 3.35.- Validación del sistema de sensores. Caso 4: Recogida de datos (incorrecto) .....	128
Tabla 3.36.- Validación del sistema de sensores. Caso 5: Calculo de información de interés (correcto).....	129
Tabla 3.37.- Validación del sistema de sensores. Caso 6: Calculo de la información de interés (incorrecto) .....	129
Tabla 3.38.- Validación del sistema de sensores. Caso 7: Envió de información al servidor (correcto) .....	129
Tabla 3.39.-Validación del sistema de sensores. Caso 8: Envió de información al servidor (incorrecto).....	129
Tabla 3.40.- Resultados obtenidos en las validaciones del sistema web .....	130
Tabla 3.41.- Resultados obtenidos en las validaciones de la aplicación móvil	131
Tabla 3.42.- Resultados obtenidos en las validaciones del sistema de sensores .....	131

# Capítulo 1

---

## 1. Introducción

### 1.1. Motivación

En la actualidad la Inteligencia Ambiental y la Computación Ubicua [1] han desarrollado multitud de paradigmas que integran sistemas de inteligencia que pueden ser utilizados en entornos para que las personas aumenten su calidad de vida gracias a la asistencia proporcionada.

La tendencia que originan estos paradigmas ha permitido analizar las actividades o tareas cotidianas de las personas haciendo uso de técnicas de computación ubicua y móvil. Cabe destacar que en los últimos años se han incrementado la existencia de teléfonos inteligentes (Smartphone) con grandes capacidades de procesamiento, conexión a internet y multitud de aplicaciones que han conseguido que estos dispositivos pasen a formar parte de nuestra vida, incluso a ser imprescindibles para nosotros.

El reconocimiento de actividades ha resultado ser un tema de investigación desafiante en los últimos años con el objetivo de conseguir que las personas mayores o dependientes puedan alcanzar una mayor calidad de vida el mayor tiempo posible.

El paradigma de Fog Computing [2] emerge en dicho contexto, enfocándose en que los datos y parte de su procesamiento se concentran en los dispositivos, en lugar de centrar todo el procesamiento en su totalidad en la nube. Es una forma de computación distribuida de proximidad, donde cada uno de los dispositivos conectados a la red puede procesar los datos y transmitir un resumen con la información de alto nivel. La gran ventaja de este paradigma es que el procesamiento no depende totalmente de un servidor en la nube, proporcionando además otras ventajas como privacidad y seguridad de los datos, ya que estos no viajan por otras redes de interconexión.

Por otro lado, hoy en día existen multitud de dispositivos de bajo consumo que nos permiten recoger datos de las actividades analizadas para su posterior procesamiento, como es el caso de los microcontroladores Arduino y su multitud

de sensores que existen para poder ser incorporados a las placas microcontroladoras.

Este trabajo fin de máster surge de la idea de que se puede utilizar las diferentes tecnologías de dispositivos de procesamiento y sensores existentes para ayudar a personas con algún grado leve de dependencia a la realización de tareas cotidianas, las cuales si no se realizan de forma correcta pueden a largo plazo suponer un riesgo para su salud.

En concreto, este trabajo fin de master se va a centrar en desarrollar dos dispositivos inteligentes para mejorar dos situaciones de salud reales: la ingesta de bebida y la ingesta de alimentos.

En primer lugar, en muchas ocasiones nos podemos encontrar con personas que debido a una enfermedad o simplemente por descuido no ingieren la cantidad mínima de agua recomendada. Esto puede suponer un alto riesgo para su salud o el agravante de otras enfermedades que puedan sufrir.

En segundo lugar, también, nos podemos encontrar el caso en que una persona se despierte a deshoras para comer, lo que supone claramente un trastorno alimenticio y riesgo para su salud.

Para ello, proponemos que dichas tareas puedan ser controladas a través de un conjunto de sensores que recojan datos de como se ha llevado a cabo dichas tareas y de un sistema web que procese toda la información para que un experto en salud pueda realizar un seguimiento de las tareas. Además, a través de una aplicación móvil podemos conseguir acercar los datos y valoraciones del experto al usuario con lo que podemos lograr una mejora diaria en la realización de estas tareas. En definitiva, podemos conseguir que todas las personas con este tipo de trastornos puedan ser monitorizadas y ayudadas en su vidas para conseguir que dispongan un mejor salud y calidad de vida.

## 1.2. Propuesta

El propósito de este trabajo fin de master es desarrollar un sistema web que permita a un experto en salud monitorizar la forma en la que múltiples usuarios realizan dos hábitos cotidianos de gran importancia para su salud

(ingesta de comida y de líquidos). Dicho sistema se complementa con una aplicación móvil que permita a los usuarios mejorar sus hábitos a través de las recomendaciones que ofrece y dos dispositivos IoT que alberga un conjunto de sensores que permite obtener los datos de los hábitos al ser desplegados en los objetos cotidianos asociados a dichos hábitos.

### 1.3. Objetivos

Los objetivos derivados de nuestra propuesta de trabajo fin de master serían los siguientes:

1. Estudio de las tecnologías necesaria para el correcto desarrollo de este proyecto.
2. Desarrollar una aplicación web que permita la monitorización de las tareas de múltiples usuarios, la gestión de dispositivos IoT y el cálculo de información de interés de los datos recogidos en la realización de las tareas monitorizadas.
3. Desarrollar una aplicación móvil que permita a los usuarios mejorar en la realización de sus tareas y además, dicha aplicación sea un instrumento que les ayude en la realización de dichas tareas.
4. Desarrollar un conjunto de dispositivos IoT de bajo coste que nos permita obtener datos en tiempo real de las tareas realizadas por los usuarios.
5. Desarrollar de forma correcta la comunicación entre los diferentes elementos que forma parte de nuestro proyecto a través de la implementación de un conjunto de servicios REST.
6. Realizar los manuales necesarios para el sistema web, la aplicación móvil y los dispositivos IoT.
7. Crear una memoria que describa el trabajo desarrollado para la realización de este trabajo fin de master.

### 1.4. Planificación temporal

Para entender de una forma más fácil e intuitiva la planificación temporal de este proyecto, se muestra la estructura de trabajo propuesta en la figura 1.1, junto con la estimación de tiempos de cada una de las tareas a desarrollar en



<b>Desarrollo de los sensores (hardware)</b>	<b>7</b>
<b>Desarrollo software</b>	
<b>Sensores</b>	
<b>Diseño</b>	<b>2</b>
<b>Implementación</b>	<b>18</b>
<b>Aplicación web</b>	
<b>Diseño</b>	<b>10</b>
<b>Implementación</b>	<b>25</b>
<b>Implementación de servicios REST</b>	<b>3</b>
<b>Aplicación Android</b>	
<b>Diseño</b>	<b>5</b>
<b>Implementación</b>	<b>10</b>
<b>Pruebas Software</b>	
<b>Corrección de posibles errores</b>	<b>5</b>
<b>Instalación del software</b>	
<b>Preparación del servidor</b>	<b>3</b>
<b>Despliegue de aplicaciones</b>	<b>3</b>
<b>Realización de la documentación</b>	
<b>Memoria</b>	<b>30</b>
<b>Manuales</b>	
<b>Manuales de usuario</b>	<b>2</b>
<b>Manuales de administrador</b>	<b>2</b>
<b>TOTAL</b>	<b>147</b>

Tabla 1.1. Planificación temporal

Como podemos observar, nuestra estimación nos indica que necesitaremos un total de 147 días para poder desarrollar este trabajo fin de máster.

#### **1.4.2. Diagrama de Gantt**

El diagrama de Gantt es una herramienta que nos permite planificar y programar tareas a lo largo de un periodo determinado de tiempo. A continuación, se adjunta el diagrama Gantt asociado a la planificación de este proyecto:

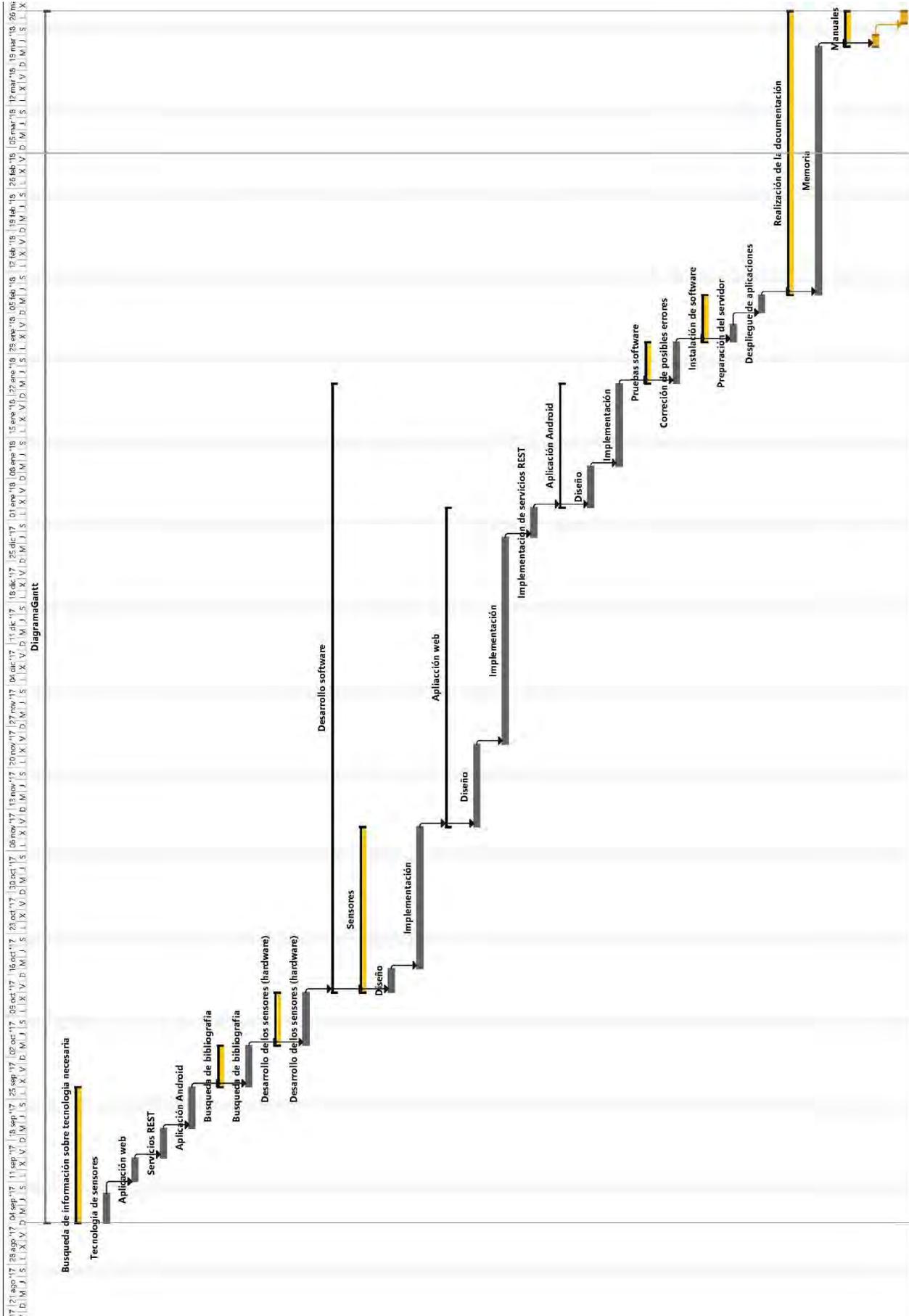


Figura 1.2.- Diagrama de Gantt

## 1.5. Presupuesto

El presupuesto de un proyecto es una de las herramientas fundamentales de la que dispone un desarrollador. Constituye la primera toma de contacto y nuestra carta de presentación de cara al cliente. En un presupuesto se intenta determinar cuanto dinero, esfuerzo, recurso y tiempo supondrá el desarrollo de un sistema o producto. En un proyecto se debe tener en cuenta los costes de materiales informáticos, dispositivos de grabación y software necesarios. Aunque no se va a incluir el resto de software utilizado para la realización de dicho proyecto ya que este será software gratuito. Por último, se incluye el coste de la mano de obra empleada para el desarrollo de dicho proyecto.

A continuación, se muestra una tabla especificando el presupuesto derivado de la realización de este proyecto.

Descripción	Precio unidad (IVA incluido)	Unidades	Coste Total (euros)
<b>Costes Software y Hardware</b>			
Alojamiento Web anual	50	1	50
Placa Lolin 32 v1.0.0	6,67	4	26,68
Sensor MPU-6050	3,58	4	14,32
Kit de Arduinio	20,10	1	20,10
Batería JST 3.7V 560mAh	6,48	4	25,92
<b>Costes de la mano de obra</b>	<i>Sueldo mensual (bruto)</i>	<i>Número de meses</i>	
<i>Análisis</i>	2541,33	1	2541,33
<i>Diseño</i>	2541,33	1	2541,33
<i>Implementación</i>	2278,83	2	4557,66
<i>Memoria</i>	2278,83	1	2278,83
<b>Coste total de la mano de obra</b>	11919,15		
<b>Coste total sin beneficio</b>	12056,17		

<b>Costes indirectos 10%</b>	1205,61
<b>Coste Total</b>	13261,78 <sup>1</sup>

Tabla 1.2.- Presupuesto

Como podemos observar, la estimación del presupuesto para realizar este trabajo fin de master nos da como resultado un valor de 13261,78 € IVA incluido.

## 1.6. Estructura de la memoria

A continuación, se describe la estructura de la memoria correspondiente a este trabajo fin de master. La memoria consta de 5 capítulos:

- **Capítulo 1:** En este capítulo, se ha explicado la motivación que nos ha llevado a desarrollar este proyecto, los objetivos a cumplir, la propuesta que debemos conseguir y finalmente, la planificación temporal de este trabajo junto con su presupuesto.
- **Capítulo 2:** En este capítulo, se detalla el hardware utilizado para la construcción de los dispositivos IoT desarrollados, los algoritmos de procesamiento y las soluciones propuestas para la monitorización de las diferentes actividades de los usuarios.
- **Capítulo 3:** Constituye un capítulo fundamental para esta memoria, al ser el capítulo dedicado al detallar el proceso completo empleado en el desarrollo de este proyecto. En primer lugar, haremos un repaso de las diferentes etapas de las que consta nuestro proceso de ingeniería software, seguido de la especificación de requisitos funcionales y no funcionales del proyecto. En segundo lugar, realizaremos el diseño de las aplicaciones desarrolladas y para finalizar, detallaremos el proceso de implementación necesario para desarrollar cada uno de los elementos de los que se compone nuestro proyecto.

<sup>1</sup> Tabla presupuesto. <http://haycanal.com/noticias/10752/los-salarios-en-informatica-y-telecomunicaciones-son-los-mas-altos-de-espana>

- **Capítulo 4:** Este es el último capítulo de la memoria y en él expondremos las conclusiones o reflexiones que nos ha aportado la realización de este proyecto y las posibles mejoras que se le puede aplicar a dicho trabajo fin de master.
- **Referencias:** Se incluyen toda la biografía utilizada a lo largo de la presente memoria.
- **Anexos:** En el Anexo A, se visualizará el contenido del CD-ROM adjunto a esta memoria, explicando con detalle cada una de las carpetas y archivos presentes a él. En los Anexos B, C y D se describen los manuales para la instalación, administración y utilización del software desarrollado en este trabajo fin de master.

## Capítulo 2

---

### 2. Introducción

Con nuestra propuesta, se intenta monitorizar el comportamiento de las personas en la realización de tareas cotidianas como puede ser comer o beber. Para monitorizar estos comportamientos vamos a desarrollar un dispositivo de bajo consumo con una placa microcontroladora que sea capaz de adaptarse a estas actividades. Es por ello, que debemos tener en cuenta que el dispositivo desarrollado para recoger datos durante las actividades cotidianas, debe ser capaz de integrarse en la propia actividad sin pasar a ser un estorbo o molestia para la persona que lo esté utilizando. Por lo que se deben optar por hardware o componentes que no supongan un problema de tamaño, peso o ergonomía por parte de la persona que lo utiliza.

Por otro lado, demos conseguir que los algoritmos de reconocimiento de actividades y procesamiento de los datos recogidos, no necesiten de una gran capacidad de cómputo ya que vamos a trabajar con una placa microcontroladora donde los recursos pueden ser limitados.

A continuación, se explica con detalle los componentes elegidos para construir nuestro dispositivo, junto con los sensores utilizados, los algoritmos de procesamiento empleados y finalmente, las soluciones propuestas para la monitorización de las actividades cotidianas (comer y beber).

#### 2.1. Arquitectura del sistema

Nuestro proyecto tal y como se ha anticipado anteriormente, consta de un sistema web que permite a los cuidadores gestionar y monitorizar las actividades de los diferente usuarios que están registrados en el mismo. Este sistema web está basado en el patrón de arquitectura Modelo-Vista-Controlador.

Nuestro sistema web se encuentra almacenado en un servidor y los cuidadores y usuarios podrán consultar la información de nuestro sistema web a través de un cliente, que en el caso de los cuidadores será un navegador web y

en el caso de los usuarios utilizaran nuestra aplicación móvil. Además, cabe destacar que nuestro sistema web ofrece una serie de servicios REST para que de este modo tanto los sensores que utilizan los usuarios en la realización de sus tareas puedan almacenar la información recogida en el sistema y los usuarios puedan a través de nuestra aplicación móvil realizar peticiones al sistema.

A continuación, se muestra una figura en que se puede ver resumida la arquitectura del sistema:

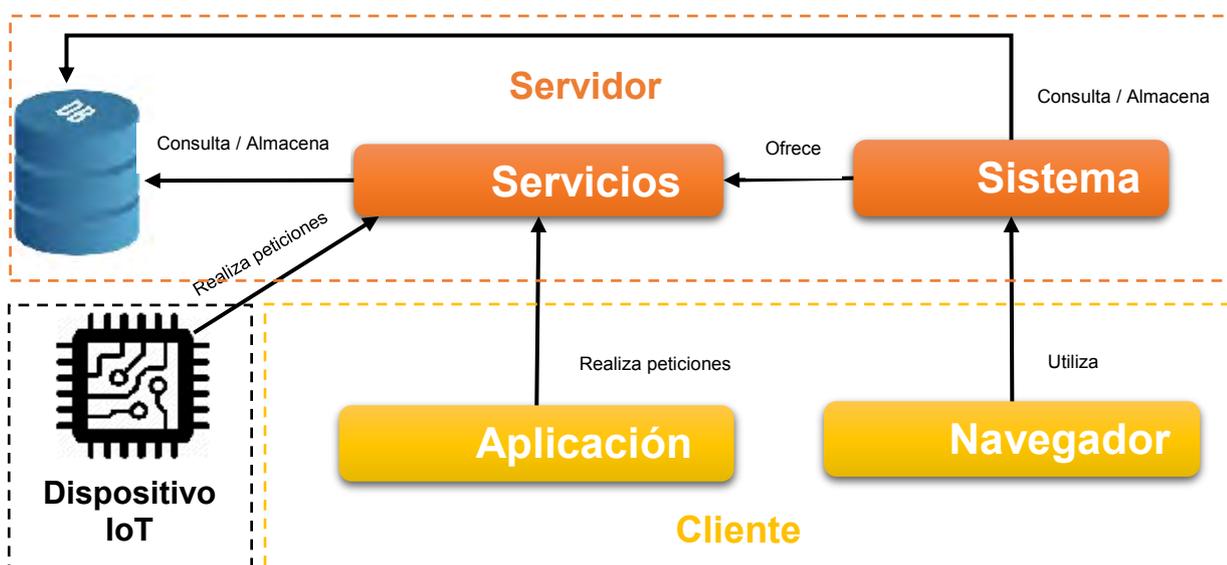


Figura 2.1.-Arquitectura del sistema

En este capítulo, se va a detallar el procesamiento de la placa microcontroladora de la que están formados nuestros sensores y en el capítulo siguiente, se procederá a explicar de formas detallada el proceso de ingeniería software empleado en el desarrollo de nuestro sistema web, aplicación móvil y software utilizado en los sensores desarrollados.

## 2.2. Descripción del dispositivo IoT

El dispositivo IoT desarrollado para poder monitorizar las tareas cotidianas (comer y beber) de múltiples usuarios debe poseer una serie de características, entre las que podemos destacar:

- Bajo consumo energético.
- Fácil instalación y utilización.
- Permitir el transporte de los datos recogidos para lo que debe soportar algún protocolo de comunicación inalámbrico (Bluetooth o Wifi).
- El sensor en la medida de lo posible debe ser transparente para el usuario y su utilización no debe ser un problema.

Todas estas características y algunas más que no se han mencionado, han influido en la elección del hardware utilizado para la construcción de nuestros sensores.

A continuación, se detalla cada uno de los componentes utilizados para construcción de nuestros sensores.

### *2.2.1. Placa*

La placa microcontroladora utilizada para la construcción de nuestro sensor es la denominada Wemos Lolin 32 que se encuentra basada en ESP-32 y cuyas características le permiten ser utilizada para este proyecto. Entre sus características podemos destacar [3]:

- Procesador basado en Espressif ESP-32 dual Core a 240 MHz
- Memoria flash de 4MB
- Wifi: 802.11 b/n/g
- Bluetooth LE
- Dimensiones: 5,8 x 2,54 cm
- Precio asequible : ~ 6€

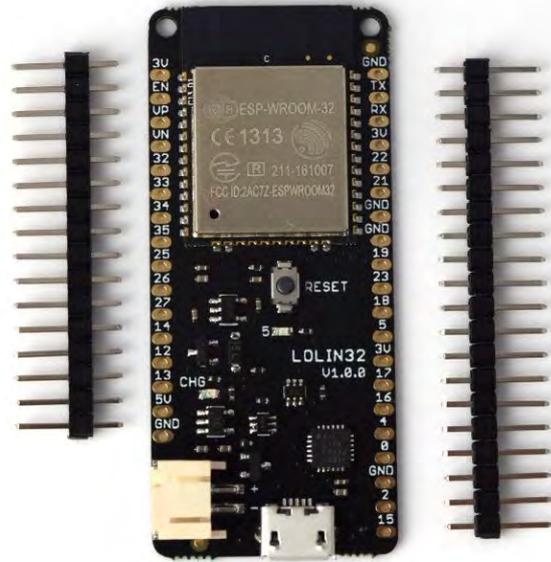


Figura 2.2.- Placa Wemos Lolin 32

### 2.2.2. Sensores

Para la monitorización de las tareas cotidianas de los usuarios, se requiere del uso del giroscopio y acelerómetro. Es por ello, que utilizamos el sensor MPU-6050 para Arduino. Este sensor contiene un acelerómetro, un giroscopio y un termómetro en un solo chip. Es muy preciso, ya que contienen hardware de conversión de analógico al digital de 16 bits por cada canal y es capaz de capturar los canales x, y, z al mismo tiempo. Además, este sensor no es caro ya que su precio esta entorno a los dos euros. [4].



Figura 2.3.-Sensor MPU-6050

### 2.2.3. Alimentación

Para la alimentación de los diferentes componentes que constituyen nuestro sensor se utilizamos una batería de Litio con las siguientes características:

- Voltaje: 3.7 V
- Capacidad: 560 mAh
- Dimensiones: 6 x 22 x 48 mm
- Conector: JST PH 2.0mm 2 pin
- Precio: ~ 6€



Figura 2.4.- Batería

### 2.2.4. Ensamblado

Para ensamblar los diferentes componentes de los que se compone nuestro sensor, se deben seguir una serie de pasos. A continuación se detallan cada uno de ellos.

En primer lugar, debemos conectar el sensor MPU-6050 a la placa microcontroladora Wemos Lolin32. Para ello, debemos seguir el esquema mostrado en la figura 2.5 y en la tabla:

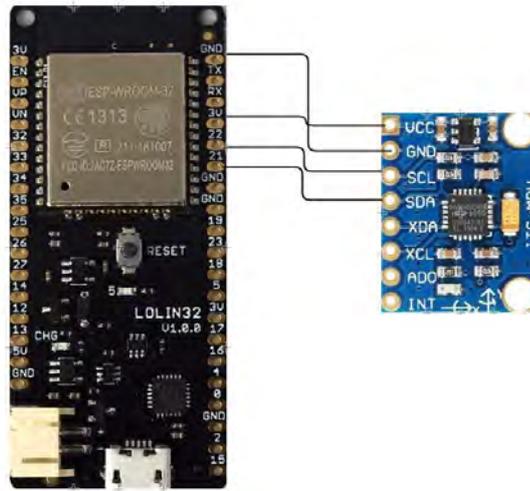


Figura 2.5.- Esquema de conexión de placa y sensor

Pin placa	Pin sensor
GND	GND
3V	VCC
22	SCL
21	SDA

Tabla 2.1.- Esquema de conexión de placa y sensor

En segundo lugar, conectamos nuestra fuente de energía al conector JST de nuestra placa Wemos Lolin32. Demos tener en cuenta que la placa utilizara la batería mientras no reciba alimentación por la conexión de 5V proporcionada por el puerto microusb. Además, este puerto microusb será utilizado para recargar la batería de Litio.



Figura 2.6.-Conector JST

Finalmente, obtendremos nuestro sensor ensamblado correctamente con todos sus componentes y listo para ser utilizado.



Figura 2.7.- Sensor ensamblado

### 2.3. Descripción de los datos recogidos

Para realizar la monitorización de las actividades de comer y beber se han recogido una serie de datos del sensor que como anteriormente se ha indicado posee un acelerómetro y giroscopio. Además, a estos datos se le ha aplicado un

pre procesamiento básico que a continuación procedemos a explicar de una forma más detallada.

En primer lugar, debemos tener en cuenta que los datos que el sensor nos proporciona no están en las unidades del sistema internacional por lo que debemos aplicar una transformación. Dicha transformación, depende de la configuración del sensor ya que en función de su configuración podemos trabajar con unos rangos u otros (para el acelerómetro: 2g/4g/8g/16g y para el giroscopio: 250/500/1000/2000(°/s)). En nuestro caso trabajamos con los rangos por defecto de 2g para el acelerómetro y 250°/s para el giroscopio [5].

Variable	valor mínimo	valor central	valor máximo
Lectura MPU6050	-32768	0	+32767
Aceleración	-2g	0g	+2g
Velocidad angular	-250°/s	0°/s	+250°/s

Figura 2.8.- Rangos del sensor MPU-6050

Una vez establecida la configuración del sensor, podemos aplicar una transformación sencilla para conseguir obtener valores en el sistema internacional. En la siguiente formular se puede observar la transformación aplicada.

$$Acelerometro_x = valor\ bruto\ acelerometro_x * \frac{9,8}{16384}$$

$$Acelerometro_y = valor\ bruto\ acelerometro_y * \frac{9,8}{16384}$$

$$Acelerometro_z = valor\ bruto\ acelerometro_z * \frac{9,8}{16384}$$

$$Giroscopio_x = valor\ bruto\ giroscopio_x * \frac{250}{32768}$$

$$Giroscopio_y = valor\ bruto\ giroscopio_y * \frac{250}{32768}$$

$$Giroscopio_z = \text{valor bruto giroscopio}_z * \frac{250}{32768}$$

En segundo lugar, eliminamos algunos valores residuales procedentes de la aceleración de la tierra (9,8 m/s<sup>2</sup>) y pequeños valores procedentes de una leve inclinación del sensor a la hora de ser colocado sobre la placa microcontroladora ( $\pm 0.5$ ).

En tercer lugar, establecemos una tasa de frecuencia de 10 medidas cada 1 segundo ya que suponemos que incrementar esta tasa de frecuencia no tendrá sentido ya que dispondríamos de muestras poco representativas.

En cuarto lugar, según la actividad que vayamos a analizar aplicamos una serie de transformaciones sobre los datos:

- En el caso de analizar la cantidad de líquido ingerida por el usuario, aplicamos el filtro complementario con el fin de calcular el ángulo de inclinación del sensor en cada una de sus componentes (x, y, z) y demás, poder eliminar posibles errores de medida [6].

Para aplicar este filtro aplicamos las siguientes formulas:

$$AnguloX_n = (AnguloX_{n-1} + GiroscopioX * \Delta t) + (\alpha * AcelerómetroX)$$

$$AnguloY_n = (AnguloY_{n-1} + GiroscopioY * \Delta t) + (\alpha * AcelerómetroY)$$

$$AnguloZ_n = (AnguloZ_{n-1} + GiroscopioZ * \Delta t) + (\alpha * AcelerómetroZ)$$

*Siendo  $\Delta t$  el valor de la tasa de frecuencia*

*Siendo el valor de  $\alpha = 0.02$*

Cabe destacar que el ángulo en que nosotros nos basaremos para calcular la cantidad de líquido ingerida por el usuario será el de la componente x ya que la posición del sensor en el vaso nos determina que nos decantemos por este ángulo.

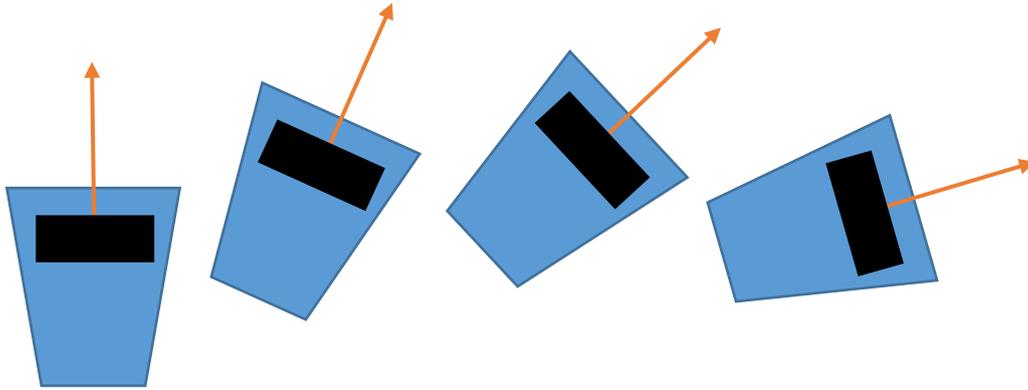


Figura 2.9.- Movimiento de beber agua

- En el caso de monitorizar la velocidad de comer, aplicamos un filtro de paso bajo de  $10 \text{ m/s}^2$  sobre los valores obtenidos en cada una de las componentes del acelerómetro para obtener los movimientos elevados que realiza el usuario y descartar el ruido.

Seguidamente, para computar la cantidad de movimiento elevado en una ventana temporal de 5 segundos que recoja la acción de comer del usuario, sumamos las últimas 50 muestras de los valores obtenidos. Estos valores fueron aconsejados por un experto de forma general, pero pueden ser adaptados a cada paciente.

En quinto lugar, debemos tener en cuenta que para el caso de analizar la velocidad de comer, el algoritmo no procederá a recoger datos y a aplicarle las transformaciones descritas anteriormente hasta que el usuario notifique que ha comenzado la actividad de comer. En el momento que el usuario comience a comer el sensor ira recogiendo datos y en intervalos de 5 segundos aplicara el algoritmo desarrollado. Con este retardo, conseguimos no agobiar al usuario con cambios rápidos de velocidad.

En cambio, en el caso de analizar la cantidad de líquido ingerida, el sensor es capaz de detectar cuando comienza el movimiento, comenzar a recoger datos, detectar cuando finaliza dicho movimiento y una vez haya finalizado, aplicar el algoritmo desarrollado. Para conseguir esto aplicaremos una serie de criterios:

- Descartar tramas de datos que no posean un número de registros mínimos ya que en las pruebas nos hemos dado cuenta de que un movimiento de beber suele generar entre 15 y 100 registros. Lo que nos permite descartar movimiento no deseados.
- Aplicamos un suavizado exponencial de valor alpha sobre la toma real leía la normal  $i_n$  de que conforman las componentes del acelerómetro. El valor estimado  $o_n$  pondera con el valor previo  $o_{n-1}$  y la toma actual  $i_n$ :

$$o_n = (o_{n-1} * alpha) + (i_n * (1 - alpha))$$

$$i_n = \left( \sqrt{Ac_{x(n)}^2 + Ac_{y(n)}^2 + Ac_{z(n)}^2} \right)$$

Para finalizar, aplicamos los algoritmos desarrollados para cada uno de los casos analizados (comer y beber) y recogemos una serie de datos de prueba para determinar la solución más aceptable para cada uno de los casos.

- **Datos para determinar la velocidad de comer:** Se recogen datos durante un periodo de tiempo para los tres tipos de movimiento (lento-medio y rápido).

Además, para eliminar ruido y movimientos irrelevantes, definimos un umbral de movimiento ( $\approx 8$ ) con filtro de paso bajo. Esto nos permite detectar que el usuario está moviendo el tenedor o el movimiento no es relevante.

Posteriormente, se calcula la cantidad de movimiento durante un intervalo de tiempo determinado  $T=50$  de forma que en cada instante  $n$ , se agregan los  $n-T$  valores anteriores, previamente filtrados.

A continuación, en la tabla 2.2 se representa una muestra de 50 valores de los datos recogidos para el movimiento lento.

Valores obtenidos			Filtro de paso bajo		
Acelerómetro X (m/s <sup>2</sup> )	Acelerómetro Y (m/s <sup>2</sup> )	Acelerómetro Z (m/s <sup>2</sup> )	Acelerómetro X (m/s <sup>2</sup> )	Acelerómetro Y (m/s <sup>2</sup> )	Acelerómetro Z (m/s <sup>2</sup> )

6.71	6.91	6.71	0	0	0
8.73	8.93	8.67	0	0	0
9.35	9.47	9.47	0	0	0
9.78	9.43	9.64	0	0	0
9.61	9.64	9.58	0	0	0
9.79	9.58	9.59	0	0	0
9.9	9.61	9.59	0	0	0
17.58	10.48	9.69	17.58	10.48	0
12.33	14.85	9.75	12.33	14.85	0
6.88	12.27	9.64	0	12.27	0
5.24	10.77	9.51	0	10.77	0
5.3	7.94	10.31	0	0	10.31
5.83	6.12	15.49	0	0	15.49
13.81	5.62	12.45	13.81	0	12.45
23.29	9.06	9.01	23.29	0	0
17.87	10.81	8.53	17.87	10.81	0
14.4	10.24	8.57	14.4	10.24	0
8.43	9.76	8.41	0	0	0
5.25	7.99	8.21	0	0	0
6.01	10.23	8.34	0	10.23	0
6.55	11.88	8.65	0	11.88	0
11.78	10.92	9.4	11.78	10.92	0
11.56	8.64	9.2	11.56	0	0
13.43	7.62	9.5	13.43	0	0
15.27	7.53	10.01	15.27	0	10.01
10.98	7.44	11.08	10.98	0	11.08
4.94	7.76	11.07	0	0	11.07
5.91	8.77	10.58	0	0	10.58
4.67	9.48	11.17	0	0	11.17
8.69	9.45	11.16	0	0	11.16
12	10.14	10.46	12	10.14	10.46
9.52	18.38	10.07	0	18.38	10.07
14.55	20.48	10.52	14.55	20.48	10.52
13.77	14.63	10.47	13.77	14.63	10.47
8.49	12.54	9.8	0	12.54	0
6.48	10.87	9.57	0	10.87	0
5.52	9.43	9.63	0	0	0
6.46	9.02	9.45	0	0	0
10.86	8.13	9.56	10.86	0	0

9.85	7.3	9.2 3	0	0	0
14.47	7.61	8.8 6	14.47	0	0
16.07	8.02	8.3 6	16.07	0	0
10.95	7.92	8.4 2	10.95	0	0
5.21	10.12	8.7 9	0	10.12	0
5.79	10.89	9.1 5	0	10.89	0
6.06	14.95	9.5 3	0	14.95	0
7.17	12.61	9.1 8	0	12.61	0
10.93	11.41	9.1 5	10.93	11.41	0
<b>Total obtenido</b>			279	262.66	144.64

Tabla 2.2.-Datos obtenidos para determinar la velocidad de comer

- **Datos para determinar la cantidad de líquido ingerida:** Se recogen valores de tres cantidades de líquido bien diferenciadas (20 ml, 45ml y 75 ml) con tres valores de presencia en el tiempo (50%, 75%, 80% y 90%) que dura el movimiento del ángulo máximo obtenido en el movimiento.
  - **Caso de 20 ml:**

Cantidad (ml)	AnguloMax(°)	Tiempo 50%	Tiempo 75%	Tiempo 80%	Tiempo 90%
20	16	10	7	7	3
20	17	11	6	5	1
20	16	6	4	4	3
20	14	13	11	10	7
20	20	19	16	13	4
20	11	11	6	6	5
20	14	15	11	9	6
20	20	13	8	8	5
20	13	12	10	8	4

20	20	20	8	6	3
20	21	19	13	12	8
20	19	12	3	2	2
20	16	9	6	6	3
20	18	19	15	15	10
20	22	11	4	3	1

Tabla 2.3.- Casos de 20ml

○ **Caso de 45 ml:**

Cantidad (ml)	AnguloMax(°)	Tiempo 50%	Tiempo 75%	Tiempo 80%	Tiempo 90%
45	34	28	21	19	16
45	32	28	18	16	9
45	42	27	17	14	10
45	40	21	11	10	5
45	40	34	24	22	14
45	38	30	21	18	6
45	35	25	15	13	10
45	36	29	20	19	13
45	44	34	21	18	14
45	32	28	19	18	15
45	41	34	23	21	16
45	36	27	18	17	7
45	40	31	20	14	6
45	37	28	18	15	11

45	36	24	15	13	8
----	----	----	----	----	---

Tabla 2.4.- Casos de 45 ml

○ **Caso de 75 ml:**

Cantidad (ml)	AnguloMax(°)	Tiempo 50%	Tiempo 75%	Tiempo 80%	Tiempo 90%
75	53	46	30	26	16
75	59	69	45	39	25
75	50	48	33	27	18
75	50	50	33	27	14
75	48	53	36	33	22
75	51	50	34	30	21
75	55	54	35	30	20
75	57	51	34	29	19
75	58	49	31	26	16
75	53	58	42	36	23
75	54	48	33	28	18
75	54	47	32	28	17
75	53	56	41	33	19
75	56	51	35	31	16
75	53	51	36	32	24

Tabla 2.5.-Casos de 75ml

## 2.4. Soluciones propuestas

Una vez hemos obtenido los datos indicados anteriormente, procedemos a valorar los resultados obtenidos y a decantarnos por la mejor solución para cada uno los casos.

A continuación, se detalla cada una de las soluciones obtenidas para cada una de las tareas analizadas.

### *2.4.1. Para la tarea de comer*

A través de los datos obtenidos en las pruebas realizadas, obtenemos la siguiente gráfica que determina la agregación de la cantidad de movimiento elevado durante 5 segundos:

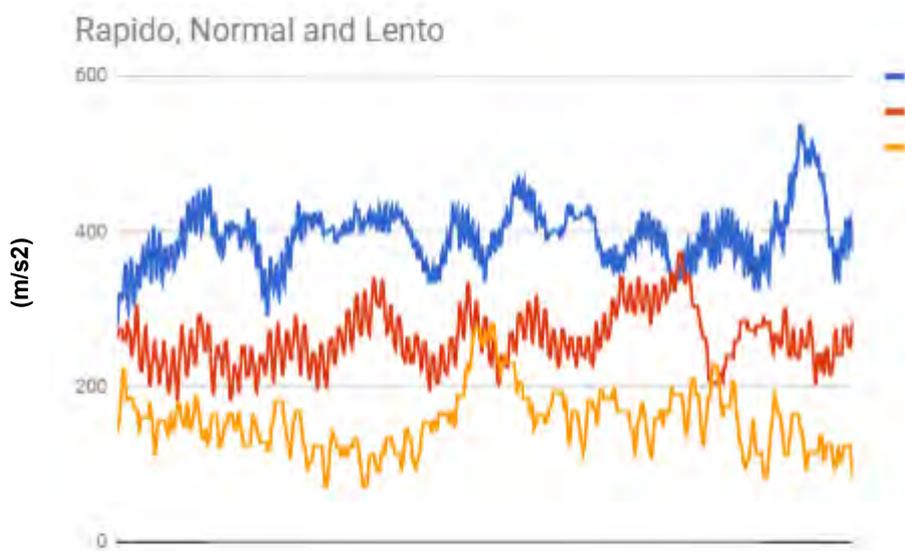


Figura 2.10.- Resultados obtenidos para la actividad de comer

En base a los datos anteriores y a la observación de un experto, se han determinado tres umbrales de movimiento:

- Un valor por debajo de 180 determina un movimiento lento
- Un valor superior a 180 e inferior a 350 determina un movimiento normal o medio.
- Un movimiento superior a 350 determina un movimiento rápido.

### *2.4.2. Para la tarea de beber*

Para conseguir alcanzar la mejor solución posible para determinar la cantidad de líquido ingerida por el usuario, haremos uso de una regresión bilineal

en la que dividiremos los datos recogidos en test y otros para el training. La regresión se ha realizado con la librería TensorFlow de Google.

La aplicación de este algoritmo nos debe obtener los valores necesarios ( $w_0$ ,  $w_1$  y  $w_2$ ) para aplicar la regresión bilineal a nuestro caso de uso.

En la siguiente tabla se muestra los resultados obtenidos para el conjunto de test:

Test	Angulo	Tiempo 50%	Tiempo 75%	Tiempo 80%	Tiempo 90%	Real	Estimación 50%	Estimación 75%
20	19	12	3	2	2	20	22.4301	18.932
20	16	9	6	6	3	20	18.2898	18.458
20	22	11	4	3	1	20	24.1808	22.338
20	14	14	9	9	4	20	19.7114	18.875
75	54	47	32	28	13	75	70.7336	71.374
75	53	56	41	33	19	75	75.3275	77.08
75	56	51	35	31	9	75	74.6886	75.355
75	53	51	36	32	16	75	72.3405	73.415
45	36	27	18	17	7	45	44.697	45.074
45	40	31	20	14	6	45	50.2174	50.104
45	37	28	18	15	11	45	46.0771	45.965
45	36	24	15	13	8	45	42.9048	42.875

Tabla 2.6.- Resultados obtenidos para la actividad de beber



Figura 2.11.- Resultados obtenidos para la actividad de beber

Además, el algoritmo empleado para la regresión bilineal nos devuelve una serie de valores de corte:  $w_0 = 0.39$ ,  $w_1 = 0.7827$  y  $w_2 = 0.5974$ :

A partir de los datos anteriores podemos decir que la estimación al 50 % es una buena solución y que la relación lineal de las componentes es un buen estimador de la cantidad de agua ingerida. Con lo que podemos estimar la cantidad de líquido ingerido a través de la siguiente fórmula:

$$\text{Cantidad ingerida} = w_0 + (\text{maxAngulo} * w_1) + (\text{Tiempo 50\%} * w_2)$$

## Capítulo 3

---

### 3. Proceso de ingeniería del software

#### 3.1. Fases de la ingeniería del software

Las fases desarrolladas en el proceso de ingeniería software para nuestra propuesta son las siguientes:

- **Especificación de requerimientos:** Consiste en realizar una descripción completa del comportamiento del sistema a desarrollar, incluyendo las funcionalidades de las que dispondrá y detallando sus requisitos.
- **Análisis del sistema:** En esta fase obtendremos un modelo del sistema claro y consistente.
- **Diseño del sistema:** Estableceremos de una forma clara y concisa el funcionamiento del sistema para satisfacer los requerimientos establecidos.
- **Implementación del sistema:** Se realiza el desarrollo del sistema descrito, teniendo en cuenta sus requerimientos y el diseño establecido.
- **Pruebas:** Se realizarán un conjunto de pruebas para verificar que el software desarrollado cumple con los requerimientos establecidos.

#### 3.2. Especificación de requerimientos

En esta fase, a través de la especificación de requerimientos o requisitos del sistema, podremos describir los servicios que debe ofrecer y las restricciones asociadas a su funcionamiento.

Esta es la primera fase dentro de nuestro proceso de ingeniería software, y es de suma importancia que se defina de forma correcta ya que es la base de nuestro software desarrollado.

Antes de especificar los requerimientos, cabe aclarar, que existen dos tipos:

- **Requerimientos funcionales:** Expresan la naturaleza del funcionamiento del sistema.
- **Requerimientos no funcionales:** Definen de forma detallada el comportamiento del sistema.

A continuación, se procede a describir con detalle cada uno de los requerimientos funcionales y no funcionales de los que se compone nuestro sistema.

### *3.2.1. Requerimientos de la aplicación web*

Seguidamente, se describen los requerimientos funcionales y no funcionales que componen nuestro sistema web

#### **3.2.1.1. Requerimientos funcionales**

Los requerimientos funcionales de nuestro sistema web serían:

1. **Login de administrador:** El sistema debe proporcionar un formulario de acceso para que los administradores del sistema puedan acceder a su panel de administración para poder así, realizar las operaciones o tareas que considere necesarias en la monitorización de sus usuarios. Además, se le debe proporcionar la funcionalidad de poder cerrar sesión para garantizar de este modo la seguridad del sistema.
2. **Crear un nuevo administrador:** El sistema debe proporcionar un formulario que permita a un administrador, crear un nuevo usuario con roles de administración.
3. **Registrar un nuevo usuario:** El sistema web debe permitir al cuidador o administrador del sistema crear un nuevo usuario en la plataforma para que de este modo se pueda realizar la monitorización o seguimiento de las tareas realizadas por el usuario. Además, debe permitir configurar las preferencias de monitorización del usuario y los objetivos que debe conseguir en la realización de sus tareas.
4. **Editar de un usuario:** El sistema debe permitir modificar los datos personales de un usuario registrado en el sistema y modificar sus objetivos o preferencias de monitorización del usuario.

5. **Control en la actividad de comer:** El sistema permitirá al cuidador llevar el control de las tareas de comer realizadas por los usuarios. Además, le mostrara datos de interés que le permitirán mejorar la monitorización de esta tarea.
6. **Control en la actividad de beber:** Nuestro sistema web permitirá al cuidador realizar de una forma fácil el seguimiento de las tareas de ingesta de líquidos por parte de los usuarios. Además, se le mostrará información relevante en la realización de dichas tareas por parte de los usuarios.
7. **Generar identificador de usuario para aplicación móvil:** El sistema web deberá ser capaz de generar un identificador único por usuario para su utilización en nuestra aplicación móvil.
8. **Avisos:** El sistema deberá ser capaz de realizar de forma automática comprobaciones en el cumplimiento de los objetivos establecidos en las tareas de los usuarios y en el caso de detectar situaciones incorrectas realizar el envío automático de avisos a nuestra aplicación móvil para que de este modo el usuario sea consciente de que debe mejorar en sus tareas.
9. **Registro de sensor:** El sistema web permitirá llevar un control de los sensores registrados en nuestro sistema y realizar acciones sobre los mismos como puede ser el caso de asociar un sensor a un usuario.
10. **Seguridad:** El sistema debe ofrecer la posibilidad de poder modificar sus credenciales de acceso a nuestro sistema y en caso de pérdida u olvido de la misma el usuario debe poder recuperar la contraseña de acceso.
11. **Contacto:** El sistema permitirá al usuario poder obtener información para poder ponerse en contacto con el equipo de desarrollo de este proyecto.

#### 3.2.1.2. Requerimientos no funcionales

Como hemos indicado anteriormente, los requerimientos no funcionales nos describirán las restricciones o limitaciones que han de tenerse en cuenta en el desarrollo de nuestro sistema.

A continuación, se detallan los requerimientos no funcionales necesarios para nuestro sistema web:

1. **Requisitos físicos:** Para entender de una forma correcta los requisitos necesarios, vamos a distinguir entre los requisitos necesarios para los usuarios que utiliza nuestro sistema web y los requisitos necesarios para el servidor donde se encuentra alojado nuestro sistema.
  - **Requisitos para el usuario:** El usuario para poder hacer uso de nuestro sistema, solo necesita disponer de un dispositivo con conexión a internet.
  - **Requisitos para el servidor:** Es necesario disponer de un servidor donde será alojado nuestro sistema web. Además, se recomienda que el servidor cumpla con los siguientes requisitos mínimos:
    - Almacenamiento de 10GB
    - RAM 1GB
    - Salida a internet
2. **Requisitos software:** Al igual que en los requerimientos físicos, vamos a dividir los requisitos software en los necesarios para el usuario y los necesarios para el servidor.
  - **Requisitos para el usuario:** El usuario para poder hacer uso de nuestro sistema, debe disponer:
    - Navegador web (Chrome, Firefox, Safari o Internet Explorer)
    - Multiplataforma gracias al desarrollo web compatible con plataformas de escritorio (Windows, Linux y Mac OS) u móvil (Android y iOS).
    - Soporte para JavaScript
  - **Requisitos para el servidor:**
    - Distribución de Linux o Windows
    - MySQL 5.5
    - Apache2
    - PHP 5.3.8
3. **Requisitos de la interfaz:** La interfaz de nuestro sistema web debe contar con los siguientes requisitos:

- **Robustez:** El sistema debe ser capaz de soportar los posibles fallos producidos por los usuarios.
- **Usabilidad:** La interfaz de nuestro sistema web debe ser intuitiva y fácil para el usuario.
- **Flexibilidad:** Debe poder adaptarse a cada uno de los dispositivos en los que el usuario desee visualizar el contenido este sistema web.

### *3.2.2. Requerimientos de la aplicación móvil*

En este apartado, vamos a detallar los requerimientos funcionales y no funcionales necesarios para nuestra aplicación móvil.

#### **3.2.2.1. Requerimientos funcionales**

Los requerimientos funcionales de nuestra aplicación móvil serán los siguientes:

1. **Login de usuario:** Nuestra aplicación móvil debe proporcionar un formulario de inicio de sesión para el usuario a través del cual se introduce un identificador y de este modo la aplicación estará vinculada a los datos del usuario.
2. **Control de la actividad de comer:** La aplicación móvil debe poseer diversas funciones que nos permitan monitorizar la actividad de comer del usuario y mostrarle información útil de las actividades realizadas. Entre las funciones podemos destacar:
  - **Control de la velocidad de comer:** Esta funcionalidad permitirá al usuario controlar su velocidad de comer en cada una de las actividades diarias en que realice dicha actividad. Para ello, cada vez que el usuario realice la acción de comer pulsara en un botón de inicio para que de este modo los sensores comiencen a recoger datos de su movimiento y se le mostrara una velocidad (lenta-media-alta). Al finalizar el usuario debe pulsar en parar para indicar de este modo que ha finalizado su actividad.

- **Histórico de actividades:** La aplicación móvil mostrará al usuario un histórico de todas las actividades de comer realizadas por el mismo y agrupadas por días. Además, se le mostrará otros datos de interés como son los horarios de comida establecidos por el cuidador o los minutos que ha pasado realizando la actividad...etc. Con esta funcionalidad se intentara que el usuario a través de estos datos pueda mejorar en sus tareas y comprobar el estado de las actividades realizadas.
3. **Control de la actividad de beber:** Nuestra aplicación móvil dispondrá de diversas funcionales que permitan al usuario llevar el control de las actividades diarias de beber líquidos. Entre estas funcionales podemos destacar:
- **Información del líquido consumido en el día:** Esta funcionalidad permitirá al usuario llevar el control de la ingesta de líquido en el día actual.
  - **Histórico de actividades:** La aplicación móvil permitirá al usuario consultar la cantidad de líquido consumidos en otros días. De este modo el usuario podrá llevar un control de calidad con la que está realizando la tarea de beber líquido.
4. **Recibir notificaciones:** La aplicación móvil debe ser capaz de recibir notificaciones enviadas por el servidor con el fin de conseguir que el usuario cumpla con los objetivos establecidos en las tareas definidas anteriormente.
5. **Cierre de sesión:** La aplicación móvil debe de permitir cerrar la sesión del usuario identificado. Además, esta funcionalidad garantiza que nuestra aplicación móvil no pueda ser utilizada por cualquier otro usuario identificado en el sistema web diferente al cuidador.

#### 3.2.2.2. Requerimientos no funcionales

Entre los requerimientos no funcionales de nuestra aplicación móvil podemos destacar:

1. **Requisitos físicos:** El usuario deberá disponer de un dispositivo móvil o Tablet con estas mínimas características:
  - Android 2.3.3 o superior
  - Conexión a internet (3G, 4G o Wifi)
  - Disponibilidad de almacenamiento en el dispositivo
2. **Requisitos de la interfaz:** La interfaz de nuestra aplicación móvil debe constar de las siguientes características:
  - **Usabilidad:** Debe disponer de una interfaz fácil e intuitiva que permita al usuario su utilización correcta.
  - **Robustez:** La interfaz debe poder soportar los posibles fallos que puedan ocurrir en el proceso de iteración con el usuario.

### *3.2.3. Requerimientos del sistema de dispositivos IoT*

En este apartado, vamos a detallar los requerimientos funcionales y no funcionales necesarios para nuestro sistema de dispositivos IoT.

#### **3.2.3.1. Requerimientos funcionales**

Entre los requerimientos funcionales de nuestro sistema a de sensores podemos destacar:

1. **Recogida de datos:** Los dispositivos IoT deberán ser capaces de realizar una recogida de datos del giroscopio y acelerómetro.
2. **Procesamiento de los datos:** El sistema deberá ser capaz de realizar tareas de procesamiento en los datos recogidos en los dispositivos IoT para que de este modo resulte más fácil aplicación los algoritmos desarrollados.
3. **Calculo de la información de interés:** En función del sensor que se trate deberá ser capaz de determinar la velocidad de movimiento según sus datos recogidos (sensor de monitorización de la tarea de comer) o calcular la cantidad de líquido ingerida a través de los datos recogidos (sensor de monitorización de la actividad de beber).

4. **Identificación de sensor:** Cada sensor dispondrá de un identificador único que será impreso en el momento de su fabricación y que será utilizado para ser asociado a un usuario.
5. **Envío de la información al servidor:** Los dispositivos IoT deberán ser capaces de realizar el envío de los datos de interés calculados al servidor central para que de este modo puedan ser utilizados por el cuidador para realizar el seguimiento o monitorización de las tareas realizadas por el usuario.

### 3.2.3.2. Requerimientos no funcionales

A continuación se detallan los requerimientos no funcionales necesarios para el sistema de dispositivos IoT desarrollado:

1. **Requisitos físicos:** Para detallar los requisitos necesarios, se han desglosado en dos bloques: requisitos físicos de usuario y los requisitos necesarios para la construcción de un sensor:
  - **Usuario:** El usuario solo necesitará disponer de una red Wifi accesible con conexión a internet
  - **Sensor:** Se necesitara una placa de Arduino con un microprocesador de bajo consumo con los siguientes requerimientos mínimos:
    - Tarjeta Wifi: 802.11 b/g/n/e/i
    - Memoria interna: ROM 448 KiB y SRAM 520 KiB
    - Procesador de 1 o 2 núcleos con frecuencia de reloj 240Mhz
    - Sensor con giroscopio y acelerómetro
2. **Requisitos software:** Los algoritmos desarrollados no deben poseer un alto coste computacional.
3. **Requisitos de usabilidad:** Nuestros dispositivos IoT deben ser integrables en los objetos cotidianos de los usuarios si afectar a su actividad diaria.

## 3.3. Análisis del sistema

A continuación, vamos a proceder a especificar de una forma detallada los casos de uso y escenarios de nuestro sistema web y aplicación móvil.

### *3.3.1. Casos de uso*

Los diagramas de casos de usos nos permiten describir los pasos o actividades que un usuario deberá realizar para poder llevar a cabo alguna tarea dentro de nuestro sistema web o aplicación móvil.

Cabe recordar que un diagrama de caso de usos se compone de los siguientes elementos:

- **Nombre del caso de uso:** Es un identificador del diagrama.
- **Actor primario:** Identifica al usuario que interactúa con el sistema.
- **Condiciones previas:** Detallan los requisitos previos que se necesitan cumplir para realizar una tarea determinada.
- **Flujo de eventos:** Los eventos que ocurrirán al realizar la acción.

Además, como podremos observar en este tipo de diagramas los casos de uso serán representados con formas de elipses, los actores con forma de personajes y las relaciones con formas de líneas o flechas.

Finalmente, en este tipo de diagramas podemos distinguir cuatro tipos de relaciones:

- **Comunicación:** Es la relación de asociación entre un actor y caso de uso. El estereotipo de este tipo de relación es: <<comunicate>>.
- **Inclusión:** Un caso de uso incorpora explícitamente el comportamiento de otro en algún lugar de su secuencia. El estereotipo de este tipo de relación es : <<include>>
- **Extensión:** Un caso de uso base incorpora implícitamente el comportamiento de otro caso de uso.
- **Generalización:** Un caso de uso incorpora explícitamente el comportamiento de otro en algún lugar de su secuencia.

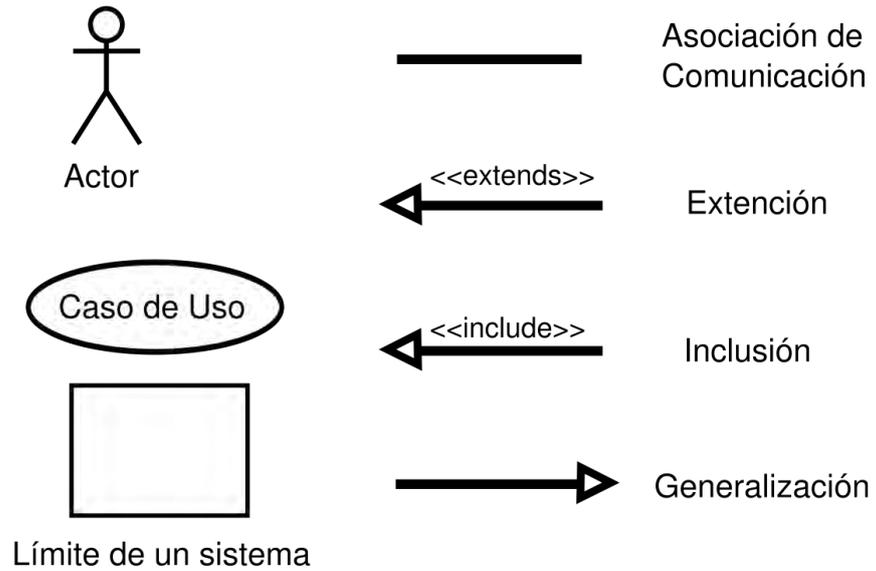


Figura 3.1.- Diagrama de casos de uso

### 3.3.1.1. Casos de uso del sistema web

En primer lugar, vamos a detallar el diagrama frontera de nuestro sistema web. Este diagrama nos permite entender de una forma fácil el funcionamiento completo de nuestro sistema.

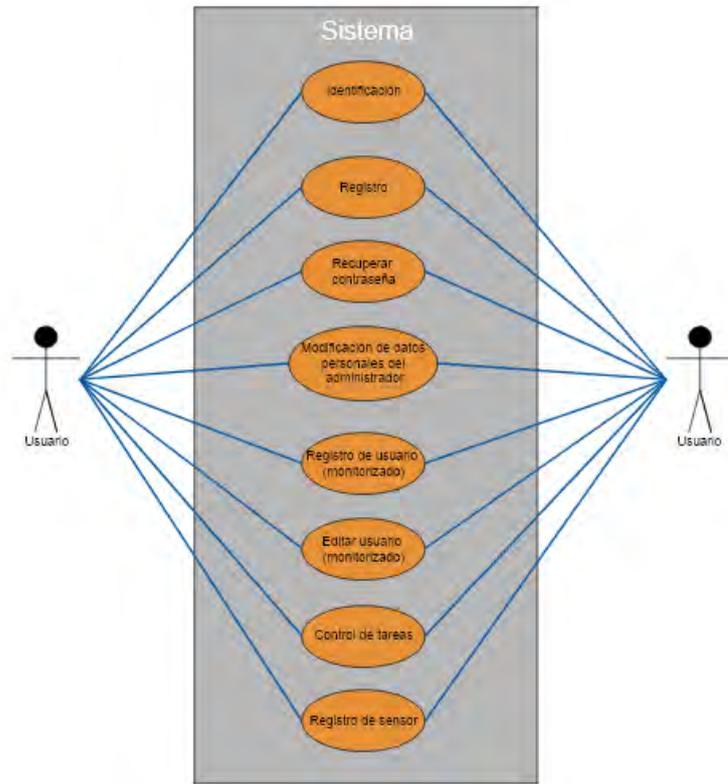


Figura 3.2.- Diagrama frontera del sistema web

Seguidamente, podremos a mostrar con más detalle los casos de uso especificados en el diagrama anterior.

### 1. Caso de uso: Identificación del administrador

- **Actuadores:** Usuario
- **Condiciones de entrada:** El usuario debe estar registrado en el sistema con permisos de administrador y en el sistema deben existir administradores
- **Eventos:**
  1. El usuario inicializa el sistema web.
  2. El sistema muestra al usuario un formulario donde puede introducir sus credenciales de acceso.
  3. El usuario introduce sus credenciales.
  4. El sistema comprueba si el usuario está registrado en el sistema y valida las credenciales introducidas.
  5. El sistema permite al usuario acceder a su panel de administración.

- **Excepciones:** El usuario introduce sus datos de acceso al sistema de forma incorrecta. El sistema pedirá al usuario que vuelva a introducir sus datos de correctamente.

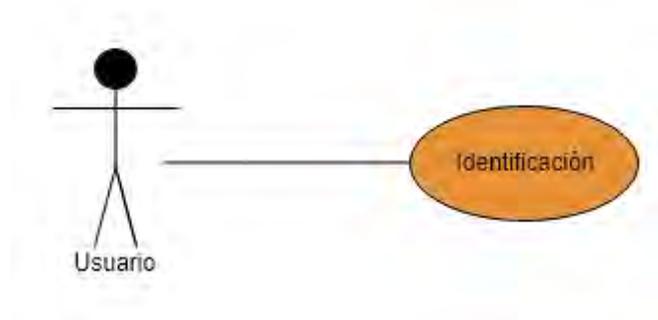


Figura 3.3.- Caso de uso: Identificación del administrador

## 2. Caso de uso: Registro de un nuevo administrador

- **Actuadores:** Usuario (administrador)
- **Condiciones de entrada:** El usuario no debe estar registrado en el sistema y otro administrador del sistema debe realizar el registro.
- **Eventos:**
  1. El administrador existente en el sistema inicia el sistema web y accede a su panel de administración.
  2. El usuario se dirige a la pestaña administradores.
  3. El usuario realiza el registro del nuevo administrador introduciendo los datos del nuevo usuario.
  4. El sistema comprueba si el usuario no está registrado en el sistema y que los datos introducidos son correctos.
  5. El sistema permite al nuevo usuario poder acceder a su panel de administración.
- **Excepciones:** El usuario introduce sus datos de acceso al sistema de forma incorrecta. El sistema pedirá al usuario que vuelva a introducir sus datos de correctamente.

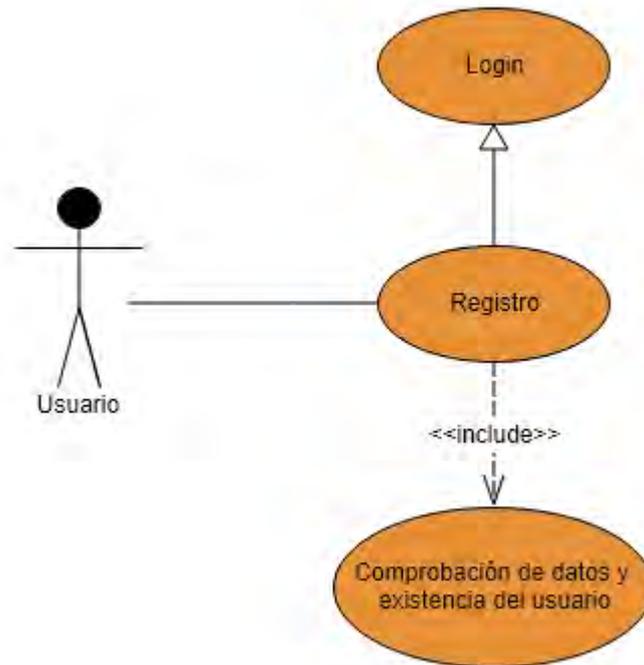


Figura 3.4.-Caso de uso: Registro de un nuevo administrador

### 3. Caso de uso: Recuperar contraseña de administrador

- **Actuadores:** Usuario (administrador)
- **Condiciones de entrada:** El usuario debe estar registrado en el sistema.
- **Eventos:**
  1. El usuario inicia el sistema web.
  2. El sistema muestra al usuario un formulario donde debe introducir sus credenciales de acceso.
  3. El usuario selecciona la opción de recordar contraseña.
  4. El sistema comprueba la existencia del usuario en el sistema web.
  5. El sistema genera de forma automática una contraseña
  6. El sistema envía al correo del usuario la nueva contraseña.
- **Excepciones:** El usuario introduce sus datos de forma incorrecta o no se encuentra registrado en el sistema. En este caso el sistema pedirá al usuario que vuelva a introducir sus datos de correctamente.

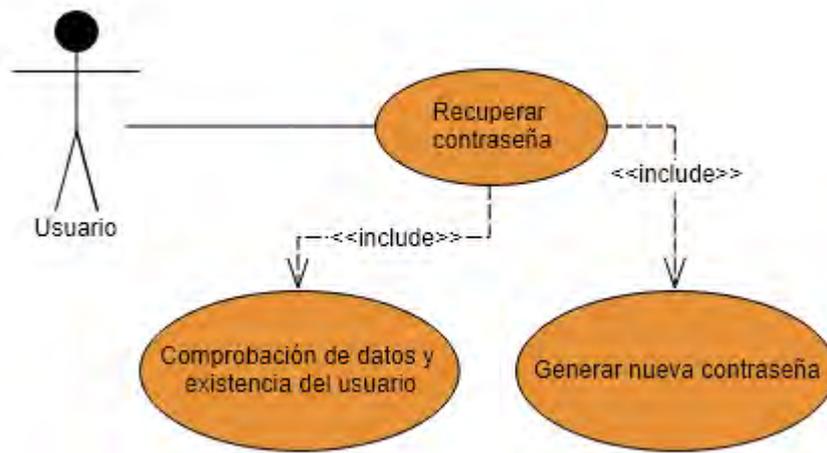


Figura 3.5.- Caso de uso: Recuperar contraseña de administrador

#### 4. Caso de uso: Modificación de datos personales del administrador (seguridad)

- **Actuadores:** Usuario (administrador)
- **Condiciones de entrada:** El usuario debe estar identificado en el sistema.
- **Eventos:**
  1. El sistema muestra al usuario un formulario con los datos personales del usuario.
  2. El usuario modifica los datos que considere necesarios y en caso de querer modificar su contraseña de acceso, el sistema le pedirá al usuario que introduzca su contraseña actual y seguidamente la nueva contraseña y una confirmación de la misma.
  3. El sistema comprueba si los datos introducidos son correctos
  4. El sistema comprueba la existencia del usuario en el sistema web.
  5. El sistema comprueba si los datos introducidos son correctos.
  6. El sistema realiza la actualización de los datos personales del usuario.
- **Excepciones:** Los datos personales introducidos por el usuario son incorrectos. En este caso el sistema le mostrará un error y pedirá de nuevo que los datos sean introducidos de forma correcta.

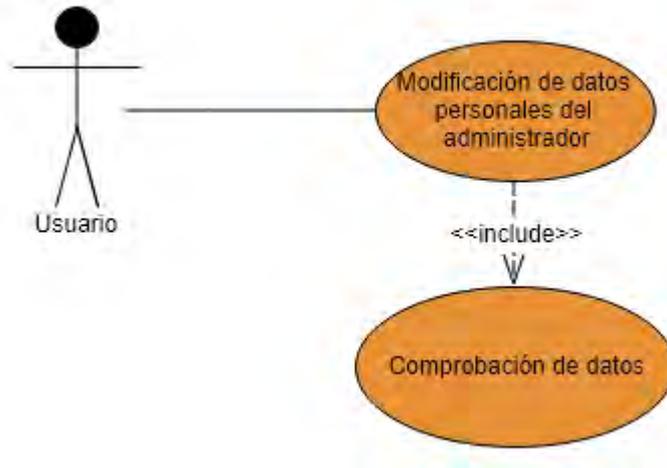


Figura 3.6.- Caso de uso: Modificación de los datos personales del administrador (seguridad)

### 5. Caso de uso: Registrar un nuevo usuario (monitorizado)

- **Actuadores:** Usuario (administrador)
- **Condiciones de entrada:** El usuario administrador debe estar identificado en el sistema. El usuario a monitorizar no debe estar registrado en el sistema.
- **Eventos:**
  1. El usuario accede a la pestaña de usuarios monitorizados dentro del panel de administración del sistema.
  2. El usuario rellena con los datos del usuario a monitorizar los datos requeridos por un formulario mostrado por el sistema para el nuevo registro.
  3. El usuario realiza una configuración inicial de preferencias y objetivos en la monitorización de las tareas desempeñadas por el usuario.
- **Excepciones:** El usuario introduce los datos de forma incorrecta. En este caso el sistema pedirá al usuario que vuelva a introducir sus datos de correctamente.

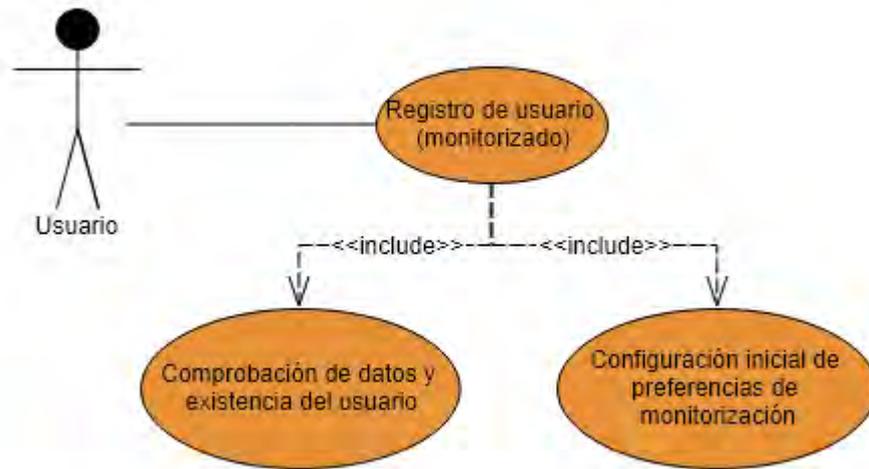


Figura 3.7.-Caso de uso: Registrar un nuevo usuario (monitorizado)

## 6. Caso de uso: Editar un usuario (monitorizado)

- **Actuadores:** Usuario (administrador)
- **Condiciones de entrada:** El usuario administrador debe estar identificado en el sistema. El usuario a editar debe estar registrado en el sistema.
- **Eventos:**
  4. El usuario accede a la pestaña de usuarios monitorizados dentro del panel de administración del sistema.
  5. El usuario selecciona al usuario monitorizado del cual quiere realizar modificaciones.
  6. El sistema muestra todos los datos sobre el usuario seleccionado.
  7. El usuario podrá modificar los datos personales del usuario seleccionado:
    - a. El usuario modifica los datos a través de un formulario mostrado por el sistema.
    - b. El usuario guarda los datos modificados.
    - c. El sistema muestra los datos del usuario actualizados.
  8. El usuario podrá modificar la configuración actual en las preferencias de monitorización del usuario seleccionado.

- a. El usuario accede a cada uno de los campos de los parámetros de monitorización (objetivos, tiempos de comida...etc.) y los modifica.
  - b. El usuario modifica los datos que considere oportunos.
  - c. El sistema muestra los datos del usuario actualizados.
- **Excepciones:** El usuario modifica los datos de forma incorrecta. En este caso el sistema mostrará un error y pedirá al usuario a realizar la acción de forma correcta.

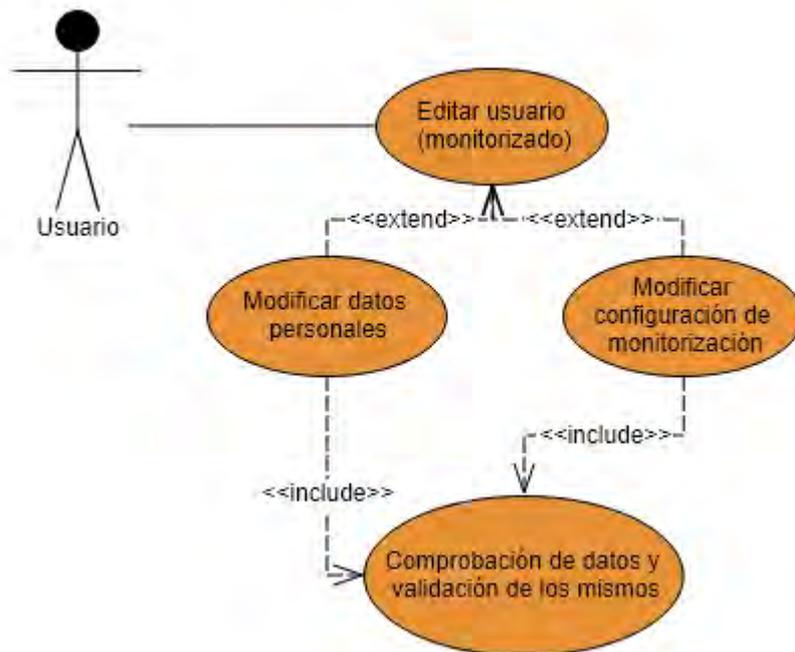


Figura 3.8.-Caso de uso: Editar un usuario (monitorizado)

### 7. Caso de uso: Control de tareas

- **Actuadores:** Usuario (administrador)
- **Condiciones de entrada:** El usuario debe estar identificado en el sistema.
- **Eventos:**

1. El usuario podrá monitorizar el estado de las tareas de comer desempeñadas por los usuarios monitorizados en el sistema.
    - a. El usuario accede a la opción de gestión de las tareas de comer dentro del panel de administración del sistema.
    - b. El sistema le muestra información sobre los datos recogidos de un usuario seleccionado de forma aleatoria.
    - c. El usuario selecciona a través de un panel de búsqueda al usuario del cual quiere comprobar el estado de sus tareas de comer.
    - d. El sistema le muestra toda la información existente en el sistema sobre el usuario seleccionado en el día actual.
    - e. El usuario a través de un formulario, podrá hacer búsquedas sobre la información recogida en días anteriores.
      - i. El sistema le mostrará la información del usuario recogida en el día seleccionado.
  2. El usuario podrá monitorizar el estado de las tareas de ingesta de líquidos de los usuarios monitorizados en el sistema.
    - a. El usuario accede a la opción de gestión de las tareas de beber dentro del panel de administración del sistema.
    - b. El sistema le muestra información sobre los datos recogidos de un usuario seleccionado de forma aleatoria.
    - c. El usuario selecciona a través de un panel de búsqueda al usuario del cual quiere comprobar el estado de sus tareas de beber.
    - d. El sistema le muestra toda la información sobre el usuario seleccionado en el día actual.
    - e. El usuario a través de un formulario, podrá hacer búsquedas sobre la información recogida en días anteriores.
      - i. El sistema le mostrará la información del usuario recogida en el día seleccionado.
- **Excepciones:** Los datos introducidos por el usuario son incorrectos, el sistema le mostrará un error y pedirá al usuario que los vuelva introducir de forma correcta.

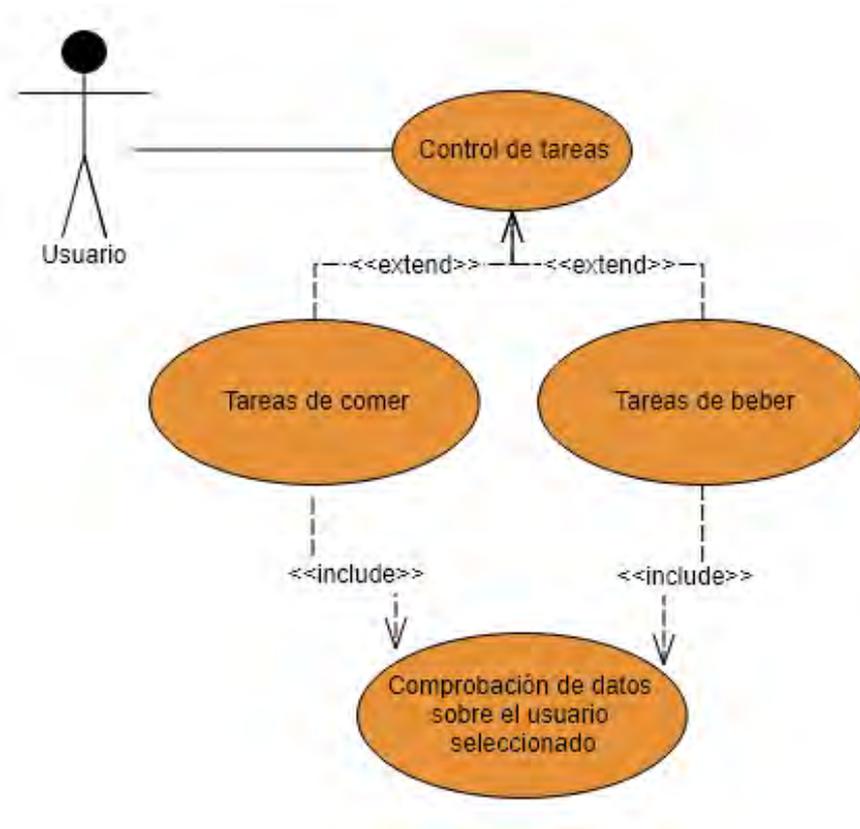


Figura 3.9.-Caso de uso: Control de tareas

### 8. Caso de uso: Registrar un nuevo sensor

- **Actuadores:** Usuario (administrador)
- **Condiciones de entrada:** El usuario administrador debe estar identificado en el sistema.
- **Eventos:**
  9. El usuario accede a la pestaña sensores dentro del panel de administración del sistema.
  10. El usuario rellena el formulario de nuevo sensor mostrado por el sistema.
  11. El usuario asocia el sensor registrado al usuario monitorizado que hará uso del sensor.
  12. El sistema muestra al usuario los datos del sensor registrado.
- **Excepciones:** El usuario introduce los datos de forma incorrecta. En este caso el sistema pedirá al usuario que vuelva a introducir sus datos de correctamente.

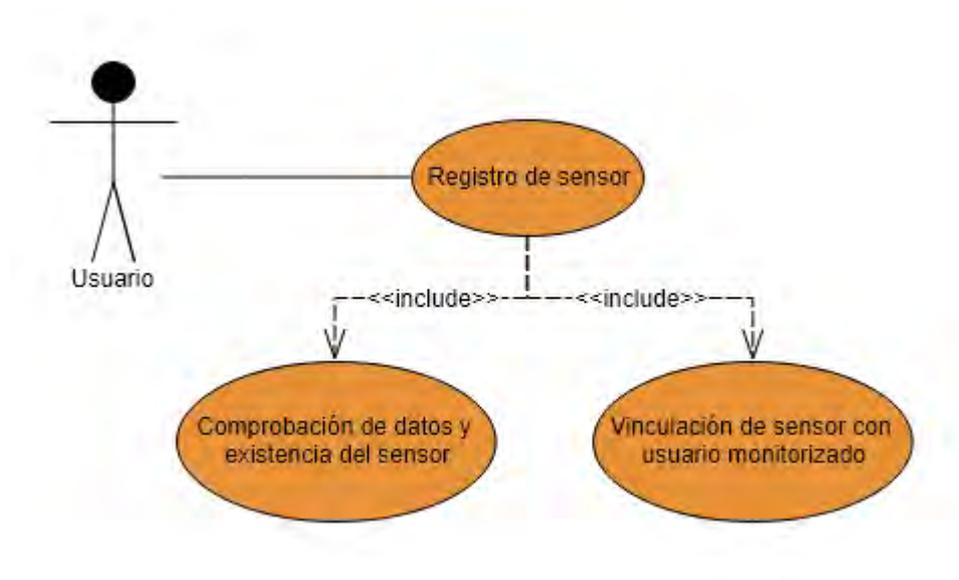


Figura 3.10.-Caso de uso: Registrar un nuevo sensor

### 3.3.1.2. Casos de uso de la aplicación móvil

A continuación, para entender el funcionamiento completo de nuestra aplicación móvil, se presenta en diagrama frontera.

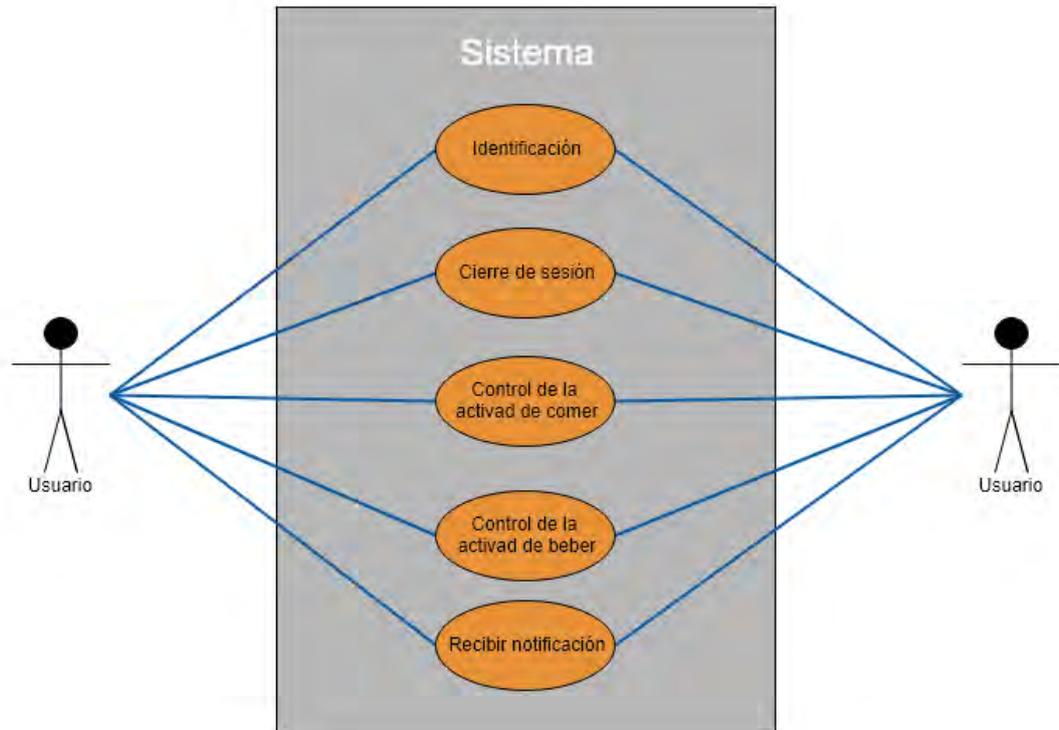


Figura 3.11.-Caso de uso: Diagrama frontera de la aplicación móvil

Seguidamente, procedemos a exponer de una forma más detallada cada uno de los casos de uso especificados anteriormente.

### 1. Caso de uso: Identificación de usuario

- **Actuadores:** Usuario
- **Condiciones de entrada:** El usuario debe estar registrado en el sistema web.
- **Eventos:**
  1. El usuario inicia nuestra aplicación móvil.
  2. El usuario rellena introduce en el formulario de acceso su identificador de sesión generado por nuestro sistema web.
  3. La aplicación móvil le muestra al usuario la pantalla principal de nuestra aplicación móvil
- **Excepciones:** El usuario introduce los datos de forma incorrecta o no se encuentra registrado en nuestro sistema web. En este caso la aplicación

pedirá al usuario que vuelva a introducir sus datos de correctamente o simplemente mostrará un error.

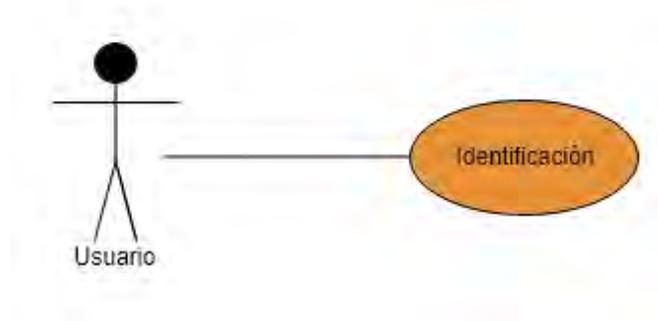


Figura 3.12.-Caso de uso: Identificar usuario

## 2. Caso de uso: Cierre de sesión

- **Actuadores:** Usuario
- **Condiciones de entrada:** El usuario debe estar identificado en nuestra aplicación móvil
- **Eventos:**
  1. El usuario inicia nuestra aplicación móvil.
  2. El usuario se dirige a la sección usuario dentro de la aplicación móvil.
  3. La aplicación móvil le muestra al usuario un mensaje de confirmación de la acción a realizar.
  4. El usuario acepta dicha acción.
  5. La aplicación móvil se cierra de forma automática.
- **Excepciones:** Durante el proceso de cierre de sesión ocurre algún problema. En este caso la aplicación móvil le mostrará un mensaje de error al usuario y se podrá continuar con la acción requerida.

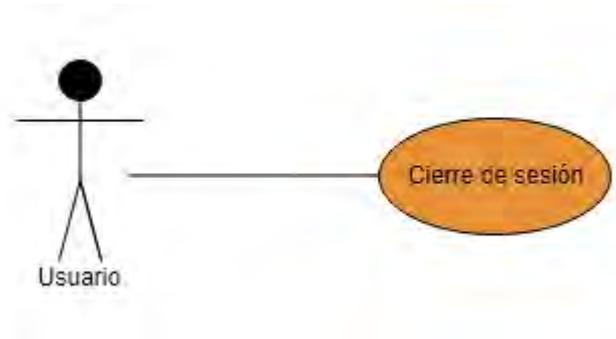


Figura 3.13.-Caso de uso: Cierre de sesión

### 3. Caso de uso: Control de la actividad de comer

- **Actuadores:** Usuario
- **Condiciones de entrada:** El usuario debe estar identificado en nuestra aplicación móvil
- **Eventos:**
  1. El usuario inicia nuestra aplicación móvil.
  2. El usuario se dirige a la sección comer dentro de la aplicación móvil.
  3. El usuario podrá controlar la velocidad con la que está realizando la acción de comer.
    - a. El usuario pulsara en el botón iniciar tarea.
    - b. La aplicación distinguirá entre tres modos de comer (lento – medio – rápido).
    - c. La aplicación móvil cambiara de forma automática la pantalla de información sobre la velocidad de comer a través de los datos ofrecidos por el sistema web que a su vez estarán basados en la información recogida por los dispositivos IoT asociados al usuario identificado en nuestra aplicación móvil.
    - d. Al finalizar la tarea de comer, el usuario pulsara en el botón finalizar tarea.
  4. El usuario podrá comprobar los datos de las tareas realizadas en días anteriores y la información de interés que le permitiría mejorar en el desempeño de sus tareas.

- a. El usuario accede a la pestaña del histórico de tareas de comer realizadas.
  - b. El usuario pulsara en el día a consultar.
  - c. La aplicación móvil le muestra al usuario la información generada según los datos recogidos en el día de la consulta.
- **Excepciones:** La aplicación móvil detecta que ha ocurrido algún error. En este caso la aplicación móvil le mostrará al usuario un mensaje alertando del error ocurrido.

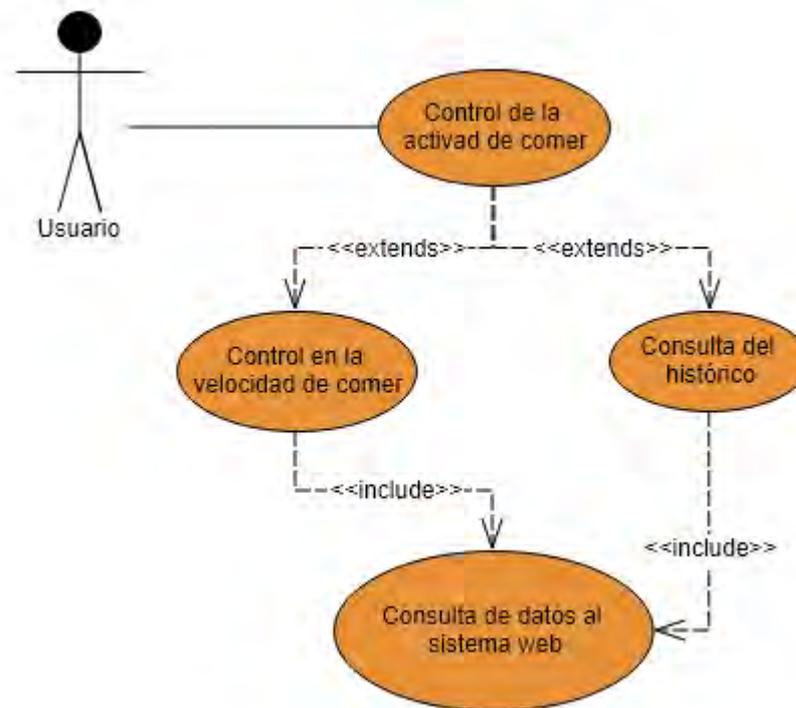


Figura 3.14.-Caso de uso: Control de la actividad de comer

#### 4. Caso de uso: Control de la actividad de beber

- **Actuadores:** Usuario
- **Condiciones de entrada:** El usuario debe estar identificado en nuestra aplicación móvil
- **Eventos:**
  5. El usuario inicia nuestra aplicación móvil.

6. El usuario se dirige a la sección beber dentro de nuestra aplicación móvil.
  7. El usuario podrá consultar el estado de su tarea de ingesta de líquidos y comprobar la cantidad ingerida en el momento de la consulta.
    - a. La aplicación móvil consultara la información requerida al sistema web y se la mostrará al usuario.
  8. El usuario podrá consultar el histórico de cantidad de líquido ingeridas en días anteriores para que de este modo pueda mejorar en el desempeño de su tarea.
    - a. El usuario accede a la pestaña del histórico de tareas de beber realizadas.
    - b. El usuario pulsara en el día a consultar.
    - c. La aplicación móvil le muestra al usuario la información generada según los datos recogidos en el día de la consulta.
- **Excepciones:** La aplicación móvil detecta que ha ocurrido algún error. En este caso la aplicación móvil le mostrará al usuario un mensaje alertando del error ocurrido.

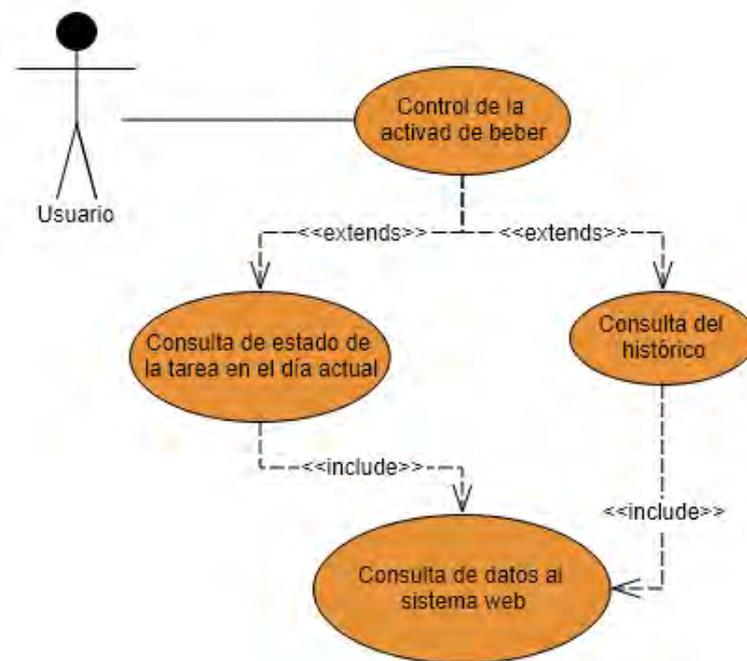


Figura 3.15.-Caso de uso: Control de la actividad de beber

### 5. Caso de uso: Recibir notificación

- **Actuadores:** Usuario
- **Condiciones de entrada:** El usuario debe estar identificado en nuestra aplicación móvil
- **Eventos:**
  1. La aplicación móvil recibe una notificación procedente del sistema web.
  2. El usuario consultara la información de la notificación recibida.
- **Excepciones:** La aplicación móvil detecta que ha ocurrido algún error. En este caso la aplicación móvil le mostrará al usuario un mensaje alertando del error ocurrido.

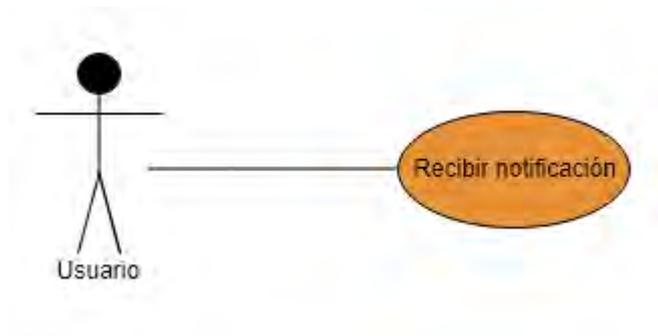


Figura 3.16.-Caso de uso: Recibir notificación

#### 3.3.1.3. Casos de uso del sistema de dispositivos IoT

De la misma forma que en los casos anteriores, primero vamos a exponer el diagrama frontera de nuestro sistema de dispositivos IoT.



Figura 3.17.-Diagrama frontera del sistema de dispositivos IoT

Seguidamente, vamos a explicar con más detalle los casos de uso que componen el diagrama frontera ilustrado anteriormente.

#### 1. Caso de uso: Recogida de datos

- **Actuadores:** Usuario
- **Condiciones de entrada:** El sensor tiene que estar vinculado a un usuario registrado en nuestro sistema web.
- **Eventos:**
  1. El usuario inicia la acción de la cual es necesario la recogida de datos.
  2. El sensor recoge los datos generados mientras el usuario realiza su tarea.
  3. El sensor realiza un procesamiento de los datos recogidos y aplica los algoritmos desarrollados para el cálculo de la información de interés en las tareas desempeñadas por el usuario.
  4. El sensor realiza el envío de los datos procesados al servidor para que de este modo puedan ser utilizados por el cuidador.
  5. Al finalizar la tarea, el sensor pasa a estado de reposo a la espera de que vuelva a producirse actividad.
- **Excepciones:** Durante el proceso de recogida de datos se detecta algún error. En este caso el sensor intentará continuar con la tarea y descartar el error generado.

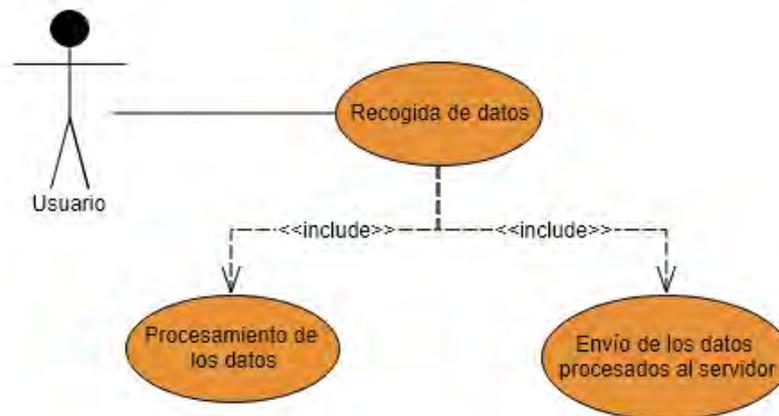


Figura 3.18.-Caso de uso: Recogida de datos

### 3.3.2. Escenarios

Podemos definir un escenario como el comportamiento del sistema ante una situación específica. La especificación de escenarios nos permite identificar distintas situaciones y describir la acción a llevar a cabo. Además, resultan de gran ayuda para identificar requerimientos.

Antes de pasar a especificar cada uno de los escenarios identificados, recordemos que un escenario se encuentra formado por un conjunto de elementos:

- Un nombre único
- Una descripción
- Los actores
- El flujo de eventos

A continuación, vamos a proceder a describir cada uno de los escenarios que conforma nuestro proyecto, para entenderlos de una manera más sencilla, vamos a diferenciar entre el sistema web, la aplicación móvil y el sistema de dispositivos IoT.

#### 3.3.2.1. Escenarios del sistema web

Los escenarios más importantes del sistema web serían:

1. **Escenario:** Identificación de administrador

- **Actores:** Antonio (administrador), Sistema web y Servidor.
- **Descripción:** Antonio quiere acceder al *panel de administración* del sistema web para comprobar el estado de las tareas realizadas por sus usuarios.
- **Flujo de eventos:**
  1. Antonio inicializa el sistema web a través de su navegador web.
  2. Antonio introduce sus *credenciales de acceso* (correo y contraseña) en el formulario de inicio de sesión.
  3. El servidor comprueba en la base de datos la veracidad de las credenciales introducidas y si son correctas, Antonio podrá acceder a su *panel de administración*.

2. **Escenario:** Registro de un nuevo administrador

- **Actores:** Antonio (administrador), Juan (nuevo administrador) Sistema web y Servidor.
- **Descripción:** Antonio quiere que Juan pueda acceder al sistema web para que pueda ayudarlo en la monitorización de las tareas de los usuarios.
- **Flujo de eventos:**
  1. Antonio inicializa el sistema web a través de su navegador web.
  2. Antonio introduce sus *credenciales de acceso* (correo y contraseña) en el formulario de inicio de sesión.
  3. El servidor comprueba en la base de datos la veracidad de las credenciales introducidas y si son correctas, Antonio podrá acceder a su panel de administración.

4. Antonio se dirige a la *pestaña administradores* del panel de administración y pulsa en la opción *añadir un nuevo administrador*.
5. Antonio rellena el formulario de registro con los datos de Juan y el sistema guarda dichos datos.
6. Antonio cierra sesión en el sistema web para que Juan inicie sesión y así poder comprobar que el registro se ha completado de forma correcta.

### 3. **Escenario:** Recuperar contraseña perdida

- **Actores:** Antonio (administrador), Sistema web y Servidor.
- **Descripción:** Antonio no recuerda la contraseña de acceso al sistema web.
- **Flujo de eventos:**
  1. Antonio inicializa el sistema web a través de su navegador web.
  2. Antonio pulsa en la opción *recordar contraseña* del panel de acceso.
  3. Antonio introduce su email y el sistema comprueba si existe una cuenta en el sistema con dicho email. Si es correcto, el sistema envía de forma automática una contraseña temporal.

### 4. **Escenario:** Registro de nuevo usuario

- **Actores:** Antonio (administrador), Juan (usuario), Sistema web y Servidor.
- **Descripción:** Antonio quiere monitorizar las actividades realizadas por Juan.
- **Flujo de eventos:**

1. Antonio inicializa el sistema web a través de su navegador web.
  2. Antonio introduce sus *credenciales de acceso* (correo y contraseña) en el formulario de inicio de sesión y accede a su *panel de administración*.
  3. Antonio se dirige a la *pestaña usuarios del panel de administración* y pulsa en el botón nuevo usuario. Seguidamente introduce los datos personales de Juan.
  4. Antonio realiza una configuración inicial de los objetivos a cumplir por Juan en la realización de las tareas.
5. **Escenario:** Editar configuración de un usuario
- **Actores:** Antonio (administrador), Juan (usuario), Sistema web y Servidor.
  - **Descripción:** Antonio quiere cambiar los objetivos que el usuario Juan debe conseguir en la realización de sus tareas de beber.
  - **Flujo de eventos:**
    1. Antonio inicializa el sistema web a través de su navegador web.
    2. Antonio introduce sus *credenciales de acceso* (correo y contraseña) en el formulario de inicio de sesión y accede a su *panel de administración*.
    3. Antonio se dirige a *la pestaña usuarios del panel de administración* y busca a Juan en el buscador de usuarios.
    4. El sistema le muestra Antonio toda la configuración realizada para el usuario Juan.
    5. Antonio realiza las modificaciones pertinentes en la configuración y guarda los cambios.

6. El sistema le muestra a Antonio la nueva configuración de Juan.

6. **Escenario:** Control de las tareas de comer

- **Actores:** Antonio (administrador), Juan (usuario), Sistema web y Servidor.
- **Descripción:** Antonio quiere acceder al *panel de administración* del sistema web para comprobar el estado de las tareas de comer realizadas por Juan.
- **Flujo de eventos:**
  1. Antonio inicializa el sistema web a través de su navegador web.
  2. Antonio introduce sus *credenciales de acceso* (correo y contraseña) en el formulario de inicio de sesión y accede al *panel de administración* del sistema web.
  3. Antonio se dirige a la pestaña comer del panel de administración y seguidamente, busca al usuario Juan en el sistema.
  4. El sistema le muestra a Antonio los datos existentes sobre Juan.

7. **Escenario:** Control de las tareas de beber

- **Actores:** Antonio (administrador), Juan (usuario), Sistema web y Servidor.
- **Descripción:** Antonio quiere acceder al *panel de administración* del sistema web para comprobar el estado de las tareas de beber realizadas por Juan.
- **Flujo de eventos:**

1. Antonio inicializa el sistema web a través de su navegador web.
2. Antonio introduce sus *credenciales de acceso* (correo y contraseña) en el formulario de inicio de sesión y accede al *panel de administración* del sistema web.
3. Antonio se dirige a la pestaña *comer* del *panel de administración* y seguidamente, busca al usuario Juan en el sistema.
4. El sistema le muestra a Antonio los datos existentes sobre Juan.

#### 8. **Escenario:** Registrar un nuevo sensor

- **Actores:** Antonio (administrador), Juan (usuario), Sistema web y Servidor.
- **Descripción:** Antonio quiere acceder al *panel de administración* del sistema web para asociar un nuevo sensor al usuario Juan.
- **Flujo de eventos:**
  1. Antonio inicializa el sistema web a través de su navegador web.
  2. Antonio introduce sus credenciales de acceso (correo y contraseña) en el formulario de inicio de sesión y accede al *panel de administración* del sistema web.
  3. Antonio se dirige a la pestaña *sensores* del *panel de administración* y pulsa en la opción *nuevo sensor*.
  4. El sistema le muestra a Antonio un formulario de registro del nuevo sensor y Antonio lo complementa de forma correcta indicando el identificador del usuario al que estará asociado dicho sensor.
  5. El sistema muestra la información del nuevo sensor.

### 3.3.2.2. Escenarios de la aplicación móvil

Los escenarios de nuestra aplicación móvil serían:

#### 1. **Escenario:** Identificación de usuario

- **Actores:** Juan (usuario), Aplicación móvil y Servidor.
- **Descripción:** Juan quiere acceder a la *pantalla principal* de la aplicación móvil.
- **Flujo de eventos:**
  1. Juan inicializa la aplicación móvil y en el formulario de identificación, introduce su identificador de usuario para nuestra aplicación.
  2. La aplicación permite a Juan acceder a la *pantalla principal*.

#### 2. **Escenario:** Cierre de sesión

- **Actores:** Juan (usuario), Aplicación móvil y Servidor.
- **Descripción:** Juan quiere cerrar su sesión en la aplicación móvil ya que ha finalizado su periodo de monitorización.
- **Flujo de eventos:**
  1. Juan inicializa la aplicación móvil y se dirige a su *perfil*. Seguidamente, pulsa en la opción *cerrar sesión*.
  2. La aplicación móvil le muestra a Juan un mensaje de confirmación sobre la tarea que va a realizar.
  3. Juan acepta la confirmación y la aplicación se reinicia.

#### 3. **Escenario:** Control de la actividad de beber

- **Actores:** Juan (usuario), Aplicación móvil y Servidor.
- **Descripción:** Juan quiere comprobar el estado de su tarea de beber.

- **Flujo de eventos:**

1. Juan inicializa la aplicación móvil y pulsa en la opción *líquido* de la pantalla principal de la aplicación móvil. Seguidamente, pulsara en la opción *Consumo actual* de la vista mostrada por la aplicación.
2. La aplicación consulta al servidor los datos solicitados por Juan y dichos datos son mostrados para que Juan pueda visualizarlos.

#### 4. **Escenario:** Consultar histórico de la actividad de beber

- **Actores:** Juan (usuario), Aplicación móvil y Servidor.
- **Descripción:** Juan quiere comprobar el estado de su tarea de beber en días anteriores.

- **Flujo de eventos:**

1. Juan inicializa la aplicación móvil y pulsa en la opción beber de la *pantalla principal* de la aplicación móvil. Seguidamente, pulsara en la opción *Historial* de la vista mostrada por la aplicación.
2. La aplicación consulta al servidor los datos solicitados por Juan y dichos datos son mostrados para que Juan pueda visualizarlos.
3. Juan selecciona los días que quiere consultar y la aplicación le muestra a Juan los datos referentes a los días solicitados.

#### 5. **Escenario:** Control de la actividad de comer

- **Actores:** Juan (usuario), Aplicación móvil y Servidor.
- **Descripción:** Juan quiere comprobar monitorizar la velocidad con la que va a comer.
- **Flujo de eventos:**

1. Juan inicializa la aplicación móvil y pulsa en la opción comer de la pantalla principal de la aplicación móvil. Seguidamente, pulsara en la opción *Monitorizar* de la vista mostrada por la aplicación.
  2. Juan pulsa en el botón *Iniciar* para indicar de este modo que va a comenzar a comer.
  3. La aplicación notifica al servidor la acción de Juan.
  4. Juan realiza movimiento mientras come y la aplicación, le va mostrando a Juan los datos procesados por los dispositivos IoT referentes a la velocidad con la que está comiendo, en base a los valores de referencia (lento-medio-rápido).
  5. Al finalizar de comer, Juan pulsa en el botón *parar* y la aplicación notificara al servidor que Juan ha finalizado su tarea de comer.
6. **Escenario:** Consultar histórico de la actividad de comer
- **Actores:** Juan (usuario), Aplicación móvil y Servidor.
  - **Descripción:** Juan quiere comprobar el estado de su tarea de comer en días anteriores.
  - **Flujo de eventos:**
    1. Juan inicializa la aplicación móvil y pulsa en la opción comer de la *pantalla principal* de la aplicación móvil. Seguidamente, pulsara en la opción *Historial* de la vista mostrada por la aplicación.
    2. La aplicación consulta al servidor los datos solicitados por Juan y dichos datos son mostrados para que Juan pueda visualizarlos.

3. Juan selecciona los días que quiere consultar y la aplicación le muestra a Juan los datos referentes a los días solicitados.

### 3.3.2.3. Escenarios del sistema de dispositivos IoT

Los escenarios de nuestro sistema de dispositivos IoT serían:

#### 1. **Escenario:** Actividad de beber

- **Actores:** Juan (usuario), Sensor y Servidor.
- **Descripción:** Juan ha iniciado su actividad de beber.
- **Flujo de eventos:**
  1. Juan procede a beber un vaso de agua.
  2. El sensor detecta que Juan ha iniciado la tarea de beber y procede a recoger datos sobre dicha tarea.
  3. Una vez que Juan se ha bebido e vaso de agua, el sensor detecta que Juan ha finalizado y calcula la cantidad de agua ingerida.
  4. El sensor envía la cantidad de agua ingerida al servidor.

#### 1. **Escenario:** Actividad de comer

- **Actores:** Juan (usuario), Sensor, Aplicación móvil y Servidor.
- **Descripción:** Juan ha iniciado su actividad de comer y quiere monitorizar la velocidad con la que está comiendo.
- **Flujo de eventos:**
  1. El sensor recibe la notificación por parte del servidor de que Juan ha iniciado su tarea de comer.
  2. El sensor comienza a recoger datos y a procesarlos para conseguir la información de interés para Juan.

3. El sensor cada cierto tiempo envía la información procesada al servidor.
4. La aplicación móvil le va mostrando a Juan los datos recogidos por el sensor referente a la velocidad de comer.
5. Juan finaliza su tarea de comer y el sensor recibe dicha notificación por parte del servidor lo que le llevará a volver a un estado de reposo a la espera de que vuelva a repetirse la actividad por parte de Juan.

### 3.4. Diseño del sistema

Una vez hemos establecidos los requisitos del software que conforman este proyecto, el diseño nos permitirá identificar los objetivos de nuestro sistema.

A continuación, para entender de una forma correcta el diseño realizado, vamos a dividirlo en tres partes, diseño de clases, diseño de datos y finalmente, diseño de la interfaz.

#### 3.4.1. Diseño de clases

En este apartado, vamos a detallar el diagrama de clases de cada una de las aplicaciones desarrolladas en nuestro proyecto.

Antes de proceder a especificar cada uno de los diagramas de clases, recordemos que un diagrama de clases es una representación gráfica que permite representar la estructura de un sistema que será implementado utilizando un lenguaje orientado a objetos. Además, un diagrama de clases está compuesto por los siguientes elementos:

- **Clase:** Contiene los atributos, métodos y visibilidad. Es la unidad básica que encapsula toda la información de un objeto.
- **Relaciones:** Puede ser de herencia, composición, agregación y uso.

También, antes de realizar el diseño de clases de nuestro sistema, hay que tener en cuenta que nuestro sistema web ha sido desarrollado bajo el patrón

MVC (modelo-vista-controlador). El objetivo de este patrón es conseguir el mayor de independencia entre el modelo de la vista y el controlador.

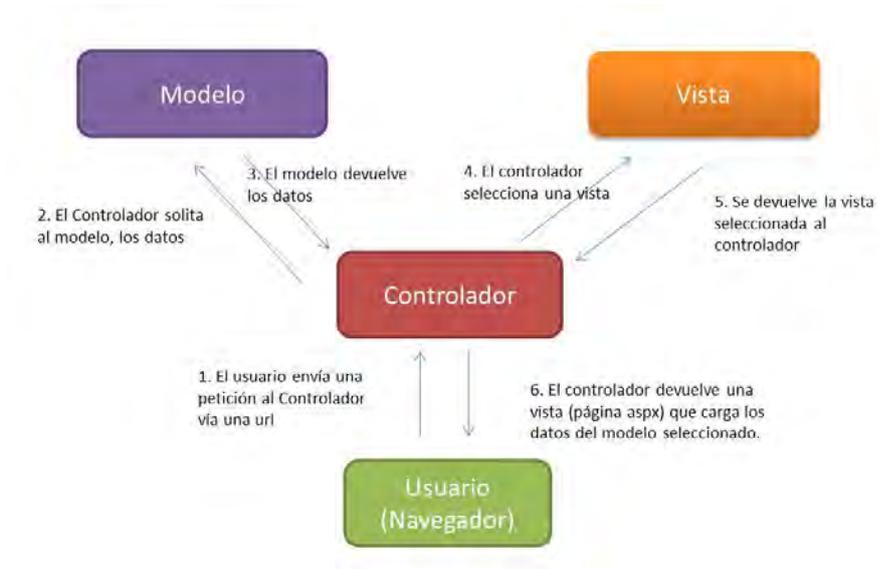


Figura 3.19.- Patrón Modelo-Vista-Controlador

Bajo este esquema, el modelo contiene todos los datos almacenados en el servidor, las vistas serán las encargadas de mostrarle los datos al cliente y el controlador será el encargado de realizar la comunicación entre la vista y el modelo. El controlador jugará un papel fundamental en el sistema ya que es el encargado de realizar su control.

Una vez introducido lo anterior, procedemos a detallar el diseño de clases de nuestro sistema web. Debido a que contamos con un gran número de clases, vamos a diferenciar cada diagrama según el controlador implicado.

- **Controlador Login:**

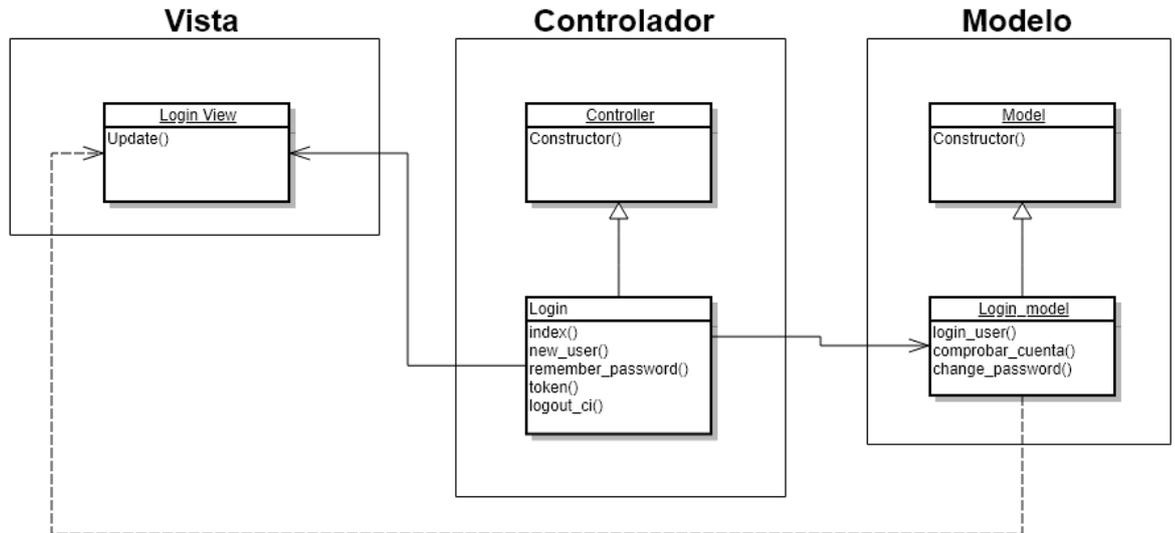


Figura 3.20.- Diseño de clases del sistema web: Controlador Login

- **Controlador “Login”**: Gestiona toda la lógica de la aplicación referente al inicio de sesión del cuidador en nuestro sistema web.
- **Modelo “Login”**: Es el encargado de resolver las solicitudes del controlador “Login” referentes a las credenciales del cuidador que intente iniciar sesión en nuestro sistema web.
- **Vista “Login\_View”**: Contiene el formulario de acceso a nuestro sistema web y el formulario de recordar credenciales.
- **Controlador Admin**:

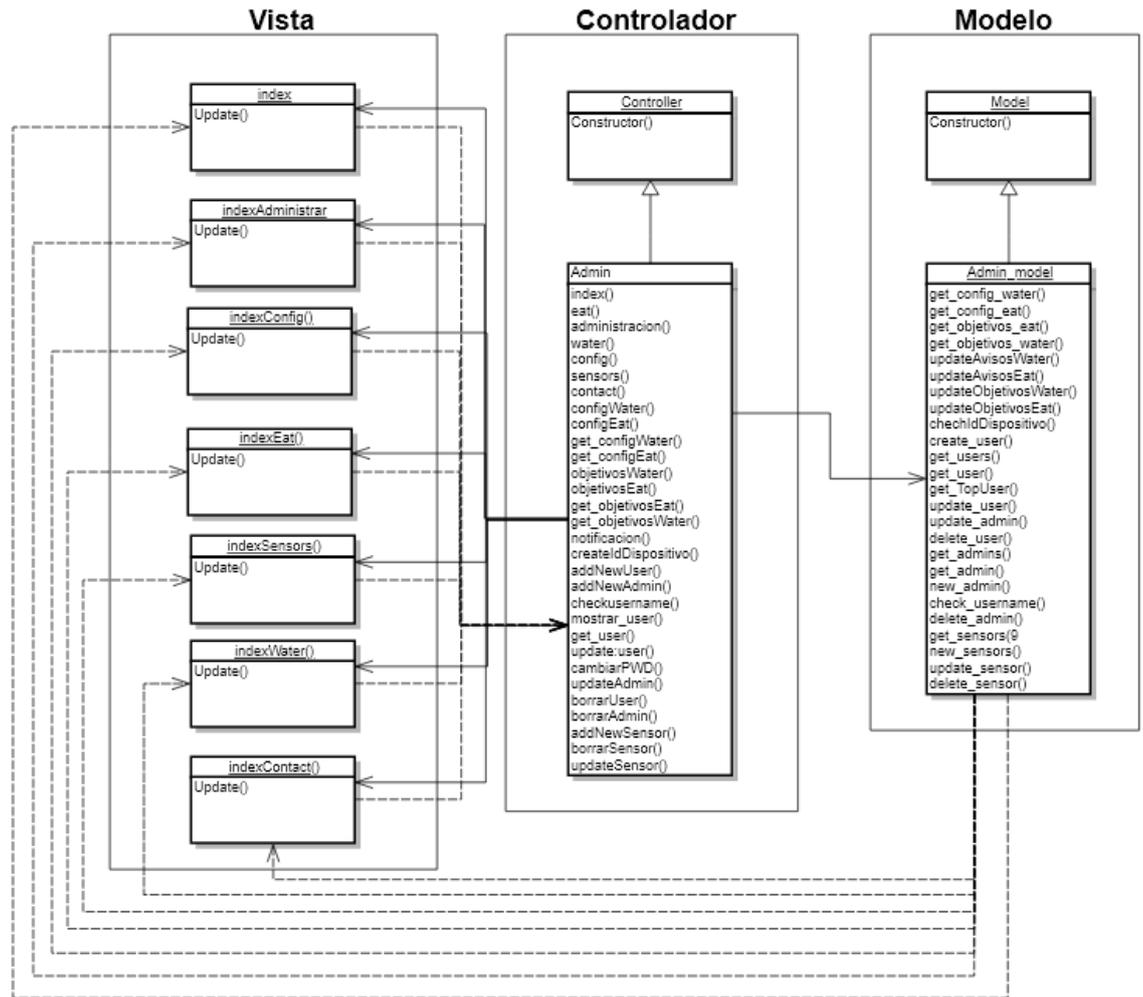


Figura 3.21.-Diseño de clases del sistema web: Controlador Admin

- **Controlador “Admin”**: Es el controlador fundamental de nuestro sistema web. Es el encargado de controlar toda la funcionalidad mostrada en el panel de administración de nuestro sistema web.
- **Modelo “Admin”**: Es el encargado de consultar la información solicitada por el controlador “Admin”.
- **Vista “index”**: Contiene la vista principal del panel de administración del sistema web.
- **Vista “indexAdministrar”**: Contiene la vista con la información referente a los administradores (cuidadores) del sistema web
- **Vista “indexConfig”**: Contiene la vista con la configuración de los objetivos para las tareas de los usuarios monitorizados.

- **Vista “indexEat”**: Esta vista muestra información referente a las tareas de comer de los usuarios.
- **Vista “indexSensors”**: Contiene información de los dispositivos IoT registrados en el sistema.
- **Vista “indexWater”**: Muestra información sobre las tareas de ingesta de líquidos de los usuarios monitorizados.
- **Vista “indexContact”**: Contiene información de contacto con los desarrolladores o responsables del sistema web.

- **Controlador Services:**

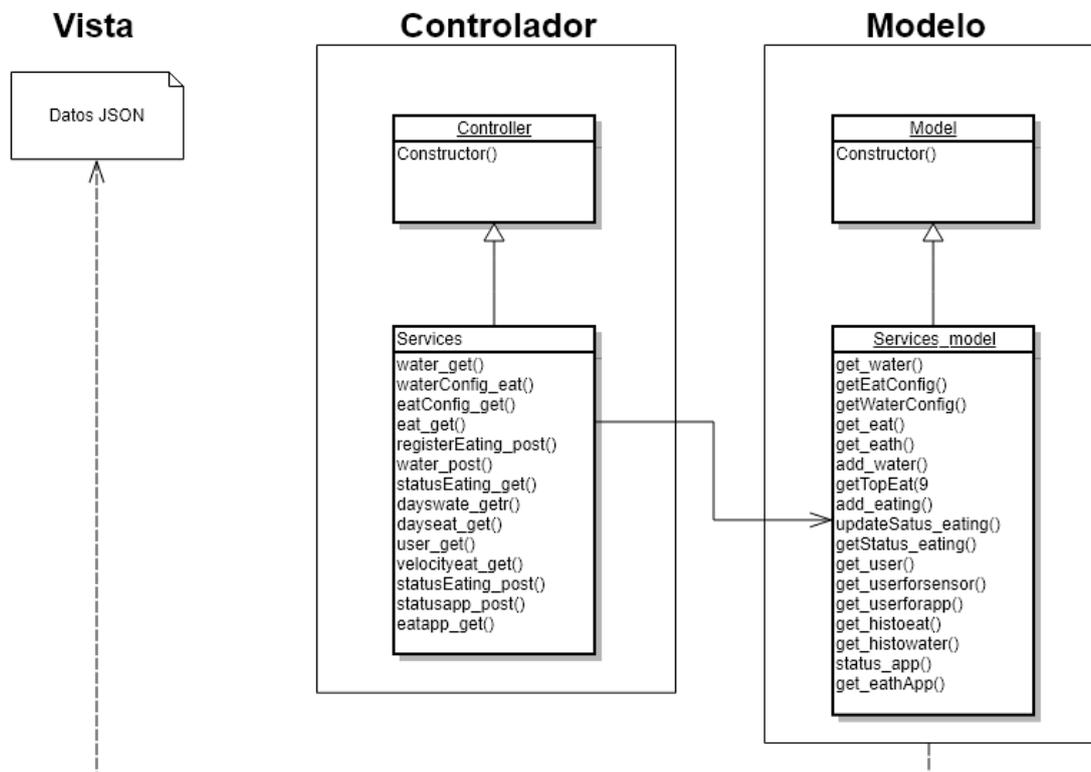


Figura 3.22.- Diseño de clases del sistema web: Controlador Services

- **Controlador “Services”**: Es el controlador encargado de ofrecer todos los servicios REST implementados en nuestro sistema web.
- **Modelo “Services\_model”**: Es el encargado de consultar la información solicitada por el controlador “Services”.
- **Vista**: Este controlador generad datos en formato JSON para que puedan ser consumidos por otros dispositivos.

Seguidamente, vamos a mostrar el diseño de correspondiente a nuestra aplicación móvil:

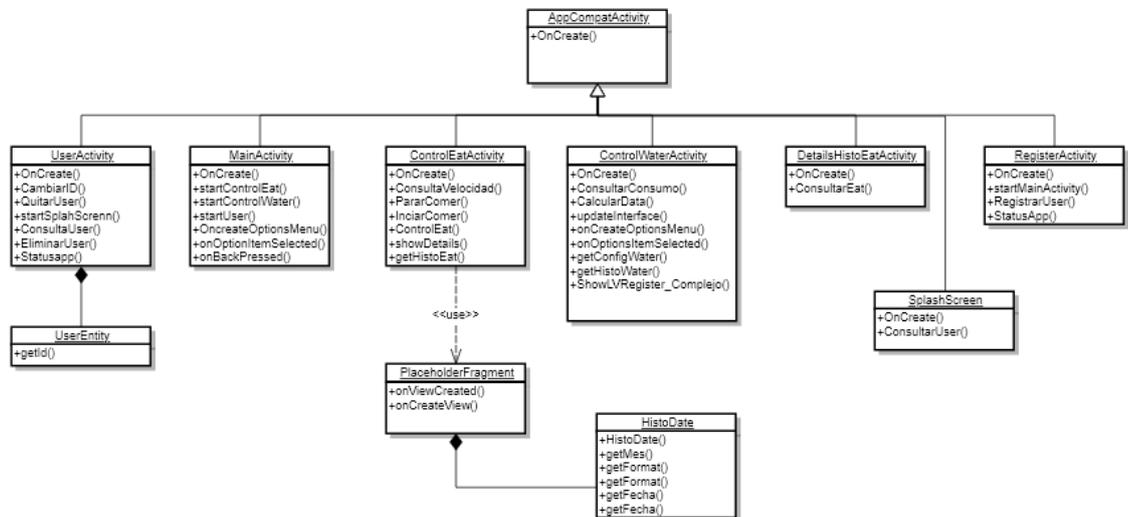


Figura 3.23.- Diseño de clases de la aplicación móvil

- **Actividad “UserActivity”**: Es la encargada de controlar la lógica de la aplicación referente a los datos del usuario y la sesión del usuario en la aplicación móvil.
- **Actividad “MainActivity”**: Controla la pantalla principal de nuestra aplicación móvil.
- **Actividad “ControlEatActivity”**: Es la encargada de controlar las tareas de comer del usuario y de mostrarle la información obtenida en la monitorización de dichas tareas.
- **Actividad “DetailsHistoEatActivity”**: Muestra al usuario información detallada sobre la realización de la tarea de comer en un día concreto.
- **Actividad “ControlWaterActivity”**: Muestra al usuario la información obtenida en la realización de las tareas de ingesta de líquidos.
- **Actividad “RegisterActivity”**: En la encargada de permitir que el usuario pueda iniciar sesión en nuestra aplicación móvil.
- **SplashScreen**: Muestra un mensaje de bienvenida al iniciar la aplicación.

- **UserEntity**: Representa la información del usuario almacenada en la base de datos local del dispositivo móvil en el cual se ejecuta nuestra aplicación móvil.
- **PlaceholderFragment**: Es el encargado de controlar el progressbar que indica al usuario la velocidad con la que está realizando la tarea de comer.

Finalmente, mostramos el diseño de clases del sistema de dispositivos IoT:

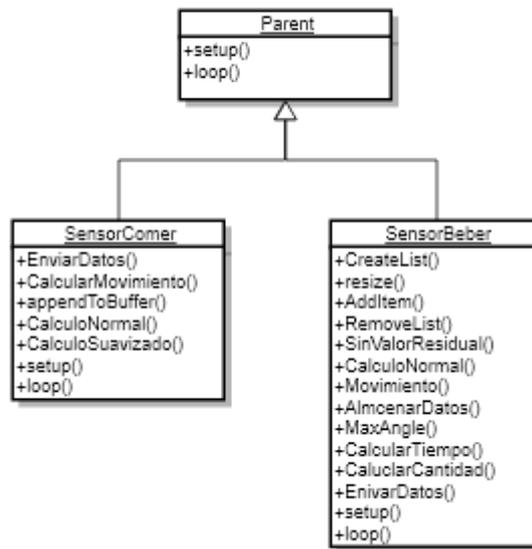


Figura 3.24.- Diseño de clases del sistema de dispositivos IoT

- **Clase “SensorComer”**: Realiza toda la lógica desarrollada para la monitorización de las tareas de comer.
- **Clase “SensorBeber”**: Realiza toda la lógica desarrollada para la monitorización de las tareas de beber.

### 3.4.2. Diseño de datos

En este apartado vamos a detallar como se encuentra estructurada la información almacenada en nuestro sistema. Para entenderlo de una forma más

fácil vamos a distinguir entre la información almacenada en nuestro sistema web y la información almacenada en nuestra aplicación móvil.

### 1. Elementos de nuestro sistema web:

- **Administradores (admins):** De cada administrador de nuestro sistema web, conoceremos su nombre, apellidos, email y sus credenciales de acceso a nuestro sistema que se encuentra compuesto de un nombre de usuario y una contraseña. Destacar que la contraseña es almacenada en modo cifrado.
- **Usuarios (users):** Este elemento identifica a los usuarios cuyas tareas son monitorizadas. De cada usuario conoceremos su nombre, apellidos, su identificador para nuestra aplicación móvil, su teléfono de contacto, su email y su estado en la aplicación móvil (Identificado y haciendo uso de nuestra aplicación móvil o no identificado y desaprovechando nuestra aplicación móvil).
- **Ingesta de líquidos (water):** Cada registro de ingesta de líquidos contendrá una marca de tiempo, la cantidad en litros consumida por el usuario y el identificador del usuario al cual pertenece dicho registro.
- **Objetivos de beber (objetivoswater):** Este elemento almacena la configuración de los objetivos que el usuario debe conseguir a la hora de realizar la tarea de ingesta de líquidos. Se almacena el identificador de usuario al cual se le aplica el objetivo y la cantidad diaria mínima de líquido que debe ingerir.
- **Tarea de comer (eat):** Almacena la marca de tiempo en el inicio de la tarea, la marca de tiempo en el que finalizó la misma, el estado de la tarea (iniciada o finalizada) y el usuario al que pertenece.
- **Registro de tarea de comer (registereat):** Este elemento contiene los datos recogidos para una tarea de comer. En dicho elemento almacenados el tipo de movimiento realizado (lento- medio- rápido) por el usuario, el valor representativo del movimiento, la marca de tiempo en el que se registró el movimiento y el identificador de la tarea de comer al cual pertenece.
- **Objetivos de comer (objetivoseat):** Este elemento contiene la configuración de un usuario para sus tareas de comer. Almacenamos los

intervalos de tiempo donde se recomienda que el usuario debe comer (desayuno, comida y cena) y el identificador del usuario al cual se le aplica dicha configuración.

- **Avisos de ingesta de líquidos (avisoswater):** Almacenamos la configuración en los avisos que el usuario recibirá en la realización de la tarea de beber. Cada registro contiene el intervalo de tiempo en el que se envían las notificaciones, cada cuanto tiempo se envía la notificación y el identificador del usuario al cual se le aplica dicha configuración.
- **Avisos de comer (avisoseat):** Se almacena los intervalos de tiempo en los que el usuario podrá recibir notificaciones y cada cuanto tiempo se le envía una notificación. Además, se almacena el identificador del usuario al cual se le aplica dicha configuración.
- **Sensores (sensors):** Almacena el identificador del sensor y el identificador del usuario al cual está vinculado dicho sensor.

## 2. Elementos de nuestra aplicación web:

- **Usuario (user):** Almacena los datos del usuario que se ha identificado en nuestra aplicación móvil. Contiene el nombre del usuario, sus apellidos, email, teléfono, identificador de la aplicación móvil y su identificador en el sistema web.

Una vez hemos especificado los elementos de los que se compone nuestro sistema web y aplicación móvil, vamos a representar en forma de tablas la base de datos asociada al conjunto de información definido anteriormente. Para ello, primero debemos realizar un diseño conceptual de la base de datos, para después obtener las tablas necesarias.

Para realizar el diseño conceptual de la base de datos vamos a utilizar el modelo Entidad-Relación (ER). Recordando un poco, un diagrama Entidad-Relación es una herramienta que nos permite representar las entidades relevantes de un sistema de información, junto con sus interrelaciones y propiedades. Además, destacar que este tipo de diagrama está formado por tres elementos principales:

- **Entidad:** Las entidades son el fundamento del modelo Entidad-Relación.

- **Atributo:** Es representado mediante un círculo o elipse y es etiquetado a través de un nombre representativo
- **Relaciones:** Se representa mediante un rombo etiquetado en su interior con un verbo.



Figura 3.25.- Diagramas Entidad-Relación

Una vez introducido el diagrama Entidad-Relación, procedemos a realizar el esquema conceptual de nuestra base de datos. Para ello vamos a diferenciar entre el esquema conceptual de nuestro sistema web y nuestra aplicación móvil.

#### 1. Esquema conceptual de la base de datos del sistema web

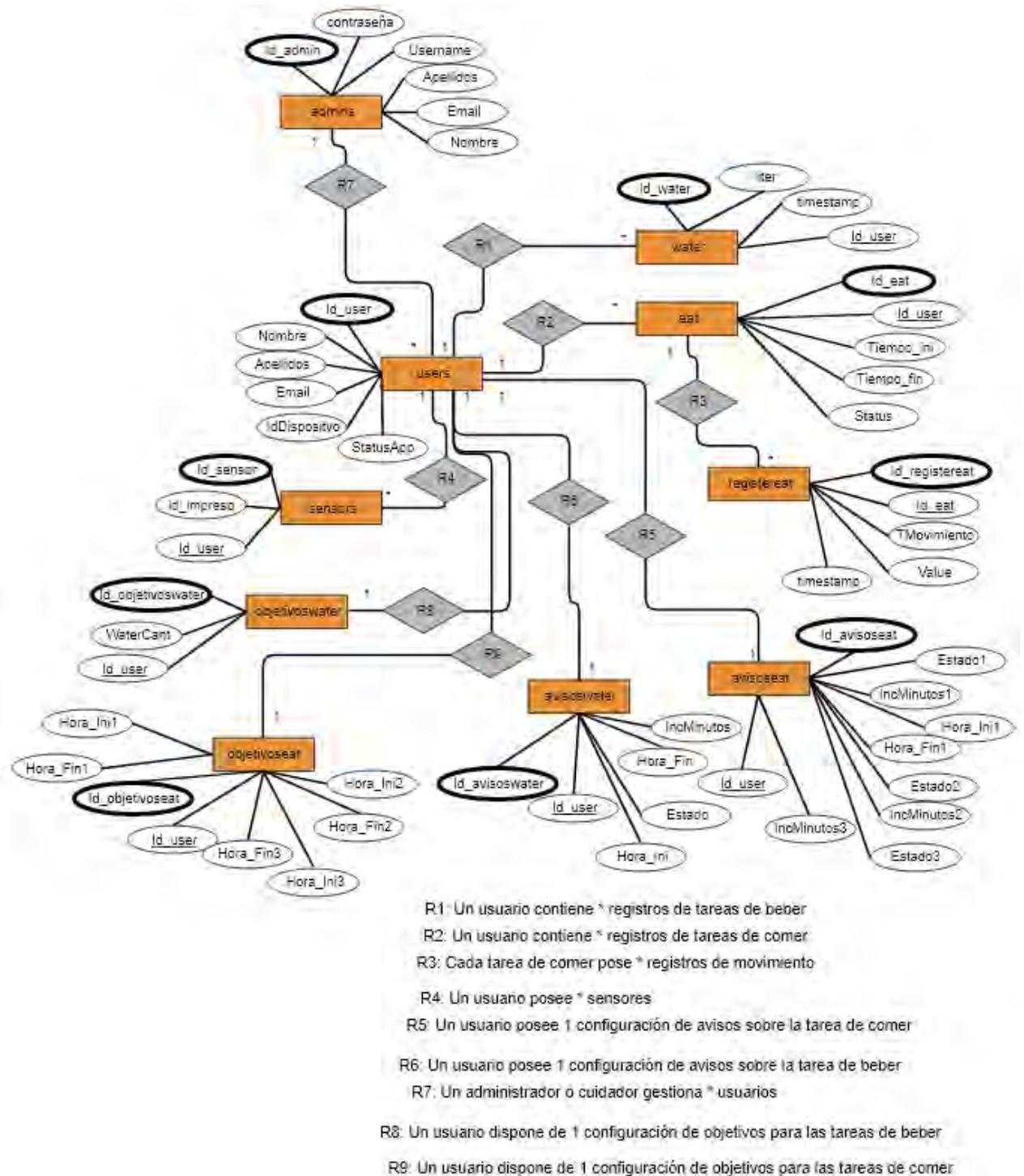


Figura 3.26.- Esquema conceptual de la base de datos del sistema web

El siguiente paso sería obtener el esquema conceptual modificado. Para ello deberíamos realizar los siguientes cambios sobre el esquema conceptual:

- Eliminar las posibles entidades débiles.

- Eliminar las relaciones de muchos a muchos.
- Eliminar todas las relaciones con los atributos existentes en el esquema conceptual.

Como podemos observar, en nuestro caso no sería necesario aplicar ninguna corrección sobre nuestro diagrama conceptual, con lo que podemos decir que nuestro diagrama conceptual coincide con el diagrama conceptual modificado.

Basándonos en el esquema obtenido, podemos definir diez tablas con una serie de atributos. A continuación, se procede a detallar con detalle cada una de las tablas que compone nuestro sistema web:

- **Admins:** Esta tabla se encuentra formada por los siguientes elementos:
  - **Id\_admin:** Clave primaria. Es un identificador único que permite identificar a cada uno de los administradores del sistema. Es un entero de 11 caracteres y es del tipo auto incremental.
  - **Nombre:** Contiene el nombre del administrador. Es una cadena de 25 caracteres.
  - **Apellidos:** Contiene los apellidos del administrador. Es una cadena de 25 caracteres.
  - **Email:** Contiene el correo electrónico del administrador. Es una cadena de 25 caracteres. Posee la restricción de no nulo.
  - **Contraseña:** Contiene la contraseña del administrador. Este campo por seguridad contiene los datos de forma cifrada y es una cadena de 100 caracteres.
- **Users:** Esta tabla está formada por los siguientes atributos:
  - **Id\_user:** Clave primaria. Es un identificador único que permite identificar a cada uno de los usuarios que componen el sistema. Es un entero de 11 caracteres y es del tipo auto incremental.
  - **Nombre:** Contiene el nombre del usuario. Es una cadena de 25 caracteres.
  - **Apellidos:** Contiene los apellidos del usuario. Es una cadena de 25 caracteres.

- **Email:** Contiene el correo electrónico del usuario. Es una cadena de 25 caracteres.
- **IdDispositivo:** Contiene el identificador único de sesión del usuario para hacer uso de la aplicación móvil. Es una cadena de 11 caracteres.
- **StatusApp:** Contiene el estado de la aplicación móvil para el usuario (iniciado sesión y haciendo uso de la misma o no conectado).
- **Water:** Esta tabla está formada por los siguientes atributos:
  - **Id\_water:** Clave primaria. Es un identificador único que permite identificar a cada una de las tareas de beber registradas en el sistema. Es un entero de 11 caracteres y es del tipo auto incremental.
  - **Liter:** Contiene el cantidad de líquido ingerida por el usuario en litros. Es de tipo double. Posee la restricción de no nulo.
  - **Timestamp:** Contiene la marca de tiempo en el que se registró la tarea. Es del tipo time. Posee la restricción de no nulo.
  - **Id\_user:** Clave foránea. Contiene el identificador del usuario que realizo la tarea. Es un entero de 11 caracteres. Posee la restricción de no nulo.
- **Eat:** Esta tabla está formada por los siguientes atributos:
  - **Id\_eat:** Clave primaria. Es un identificador único que permite identificar a cada una de las tareas de comer registradas en el sistema. Es un entero de 11 caracteres y es del tipo auto incremental.
  - **Tiempo\_Ini:** Almacena la marca de tiempo en la que se inició la tarea. Es de tipo time. Posee la restricción de no nulo.
  - **Tiempo\_Fin:** Contiene la marca de tiempo en la que finalizo la tarea. Es del tipo time.
  - **Status:** Contiene el estado de la tarea (0 finalizada, 1 iniciada). Es de tipo boolean y posee la restricción de no nulo.
  - **Id\_user:** Clave foránea. Contiene el identificador del usuario que realizo la tarea. Es un entero de 11 caracteres. Posee la restricción de no nulo.

- **Registereat:** Esta tabla está formada por los siguientes atributos:
  - **Id\_registereat:** Clave primaria. Es un identificador único que permite identificar a cada una de las tareas de comer registradas en el sistema. Es un entero de 11 caracteres y es del tipo auto incremental.
  - **TMoviento:** Contiene el tipo de movimiento detectado Es de tipo enumerado (Lento-Medio-Rápido). Posee la restricción de no nulo.
  - **Value:** Contiene el valor numérico del movimiento. Es un entero de 11 caracteres y posee la restricción de no nulo.
  - **Id\_eat:** Clave foránea. Contiene el identificador de la tarea de comer a la cual pertenece el registro. Es un entero de 11 caracteres. Posee la restricción de no nulo
  
- **Avisoswater:** Esta tabla está formada por los siguientes atributos:
  - **Id\_avisoswater:** Clave primaria. Es un identificador único que permite identificar a cada una de las configuraciones de avisos sobre las tareas de beber registradas en el sistema. Es un entero de 11 caracteres y es del tipo auto incremental.
  - **Estado:** Contiene el estado del aviso (0 no activo y 1 activo). Es del tipo boolean y posee la restricción de no nulo.
  - **Tiempo\_ini:** Contiene la hora de inicio a partir de la cual se empieza a enviar avisos. Es de tipo time. Posee la restricción de no nulo.
  - **Tiempo\_fin:** Contiene la hora de fin a partir de la cual se no se envían avisos. Es de tipo time. Posee la restricción de no nulo.
  - **IncMinutos:** Contiene el intervalo de tiempo en que se repite el aviso. Es un entero de 4 caracteres. Posee la restricción de no nulo.
  - **Id\_user:** Clave foránea. Contiene el identificador del usuario al cual se le aplica la configuración. Es un entero de 11 caracteres. Posee la restricción de no nulo.
  
- **Avisoseat:** Esta tabla está formada por los siguientes atributos:
  - **Id\_avisoseat:** Clave primaria. Es un identificador único que permite identificar a cada una de las configuraciones de avisos sobre las tareas de comer registradas en el sistema. Es un entero de 11 caracteres y es del tipo auto incremental.

- **Estado1:** Contiene el estado del aviso 1 (0 no activo y 1 activo). Es del tipo boolean y posee la restricción de no nulo.
- **Tiempo\_ini1:** Contiene la hora de inicio a partir de la cual se empieza a enviar avisos para el aviso 1. Es de tipo time. Posee la restricción de no nulo.
- **Tiempo\_fin1:** Contiene la hora de fin a partir de la cual se no se envían avisos para el aviso 1. Es de tipo time. Posee la restricción de no nulo.
- **IncMinutos1:** Contiene el intervalo de tiempo en que se repite el aviso 1. Es un entero de 4 caracteres. Posee la restricción de no nulo.
- **Estado2:** Contiene el estado del aviso 2 (0 no activo y 1 activo). Es del tipo boolean y posee la restricción de no nulo.
- **IncMinutos2:** Contiene el intervalo de tiempo en que se repite el aviso 2. Es un entero de 4 caracteres. Posee la restricción de no nulo.
- **Estado3:** Contiene el estado del aviso 3 (0 no activo y 1 activo). Es del tipo boolean y posee la restricción de no nulo.
- **IncMinutos3:** Contiene el intervalo de tiempo en que se repite el aviso 3. Es un entero de 4 caracteres. Posee la restricción de no nulo.
- **Id\_user:** Clave foránea. Contiene el identificador del usuario al cual se le aplica la configuración. Es un entero de 11 caracteres. Posee la restricción de no nulo.
- **Objetivoseat:** Esta tabla está formada por los siguientes atributos:
  - **Id\_objetivoseat:** Clave primaria. Es un identificador único que permite identificar a cada una de las configuraciones de objetivos sobre las tareas de comer registradas en el sistema. Es un entero de 11 caracteres y es del tipo auto incremental.
  - **Tiempo\_ini1:** Contiene la hora de inicio del desayuno. Es de tipo time. Posee la restricción de no nulo.
  - **Tiempo\_fin1:** Contiene la hora de fin del desayuno. Es de tipo time. Posee la restricción de no nulo.

- **Tiempo\_ini2:** Contiene la hora de inicio del almuerzo. Es de tipo time. Posee la restricción de no nulo.
- **Tiempo\_fin2:** Contiene la hora de fin del almuerzo. Es de tipo time. Posee la restricción de no nulo.
- **Tiempo\_ini3:** Contiene la hora de inicio de la cena. Es de tipo time. Posee la restricción de no nulo.
- **Tiempo\_fin3:** Contiene la hora de fin de la cena. Es de tipo time. Posee la restricción de no nulo.
- **Id\_user:** Clave foránea. Contiene el identificador del usuario al cual se le aplica la configuración. Es un entero de 11 caracteres. Posee la restricción de no nulo.
- **Objetivoswater:** Esta tabla está formada por los siguientes atributos:
  - **Id\_objetivoswater:** Clave primaria. Es un identificador único que permite identificar a cada una de las configuraciones de objetivos sobre las tareas de beber registradas en el sistema. Es un entero de 11 caracteres y es del tipo auto incremental.
  - **WaterCant:** Contiene la cantidad de líquido diaria mínima que el usuario debe ingerir. Es un entero de dos caracteres. Posee la restricción de no nulo.
  - **Id\_user:** Clave foránea. Contiene el identificador del usuario al cual se le aplica la configuración. Es un entero de 11 caracteres. Posee la restricción de no nulo.
- **Sensors:** Esta tabla está formada por los siguientes registros:
  - **Id\_sensor:** Clave primaria. Es un identificador único que permite identificar a cada uno de los sensores registrados en el sistema. Es un entero de 11 caracteres y es del tipo auto incremental.
  - **Id\_impreso:** Contiene el identificador del sensor que se le añadió en su momento de fabricación. Es un entero de 11 caracteres. Posee la restricción de no nulo.
  - **Id\_user:** Clave foránea. Contiene el identificador del usuario que hace uso del sensor almacenado en el registro. Es un entero de 11 caracteres. Posee la restricción de no nulo.

## 2. Esquema conceptual de la base de datos de la aplicación móvil



Figura 3.27.- Esquema conceptual de la aplicación móvil

Del mismo modo que en caso anterior, nuestro diagrama conceptual coincide con el diagrama conceptual modificado. Basándonos en el diagrama obtenido podemos concluir que la base de datos de nuestra aplicación móvil contiene una tabla. A continuación, se detalla los elementos que componen dicha tabla:

- **User:** Esta tabla está formada por los siguientes atributos:
  - **Id\_user:** Clave primaria. Es un identificador único que permite identificar a cada uno de los usuarios que componen el sistema. Es un entero de 11 caracteres y es del tipo auto incremental.
  - **Nombre:** Contiene el nombre del usuario. Es una cadena de 25 caracteres.
  - **Apellidos:** Contiene los apellidos del usuario. Es una cadena de 25 caracteres.
  - **Email:** Contiene el correo electrónico del usuario. Es una cadena de 25 caracteres.
  - **IdDispositivo:** Contiene el identificador único de sesión del usuario para hacer uso de la aplicación móvil. Es una cadena de 11 caracteres.

### *3.4.3. Diseño de la interfaz*

La interfaz de usuario es un aspecto a destacar ya que en función de su calidad puede conducir al sistema a su éxito o al fracaso.

A continuación, se detallaran los elementos de nuestro sistema con los que el usuario interactuará: metáforas, estilo y mensajes

#### 3.4.3.1. Metáforas

Consiste en la aplicación de un concepto o expresión sobre una idea u objeto el cual no es descrito de manera directa, con la intención de que sea comparado con otro elemento y facilitar así su comprensión.

La mayoría de metáforas que se han utilizado en el sistema web y aplicación móvil son representadas en formas de iconos que permiten hacer la interfaz más intuitiva y sencilla.

A continuación, se detallan algunas de las metáforas empeladas en el sistema web y aplicación móvil:

- **Sistema web:**
  - **Logo del sistema:**



Figura 3.28.- Logo del sistema web

Esta metáfora permite al administrador reconocer el sistema web.

- **Iconos del menú de administración:**
  - **Tarea de beber:**



Figura 3.29.- Icono de la tarea de beber

Esta metáfora permite al administrador acceder a los datos registrados por los usuarios en la realización de sus tareas de beber.

- **Tarea de comer:**

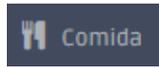


Figura 3.30.- Icono de la tarea de comer

Esta metáfora permite al administrador acceder a los datos registrados por los usuarios en la realización de sus tareas de comer.

- **Dispositivos IoT:**

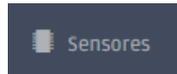


Figura 3.31.- Icono de dispositivos IoT

Esta metáfora permite al administrador acceder a los dispositivos IoT registrados en el sistema.

- **Aplicación Móvil:**
  - **Logo de la aplicación:**



Figura 3.32.- Logo de la aplicación móvil

Esta metáfora permite al usuario reconocer nuestra aplicación móvil.

○ **Perfil de usuario:**



Figura 3.33.- Icono de perfil de usuario

Esta metáfora permite al usuario acceder a su perfil y consultar sus datos personales.

○ **Información:**



Figura 3.34.- Icono de información

Esta metáfora permite al usuario obtener información sobre la aplicación móvil.

○ Valoración de la actividad de comer



Figura 3.35.- Iconos de valoración de la actividad de comer

Este conjunto de metáforas permite que el usuario pueda obtener de una forma rápida y sencilla una valoración sobre la realización de su tarea de comer (Buena o mala).

**3.4.3.2. Estilo**

En este apartado se va a especificar una guía de estilo que deben seguir todos los elementos de nuestra interfaz para que tengan coherencia y permite mostrar armonía entre las distintas aplicaciones desarrolladas en este proyecto.

A continuación, se detallan algunas reglas establecidas por nuestra guía de estilo:

- **Tipo de letra:**
  - **Sistema web:**
    - **Tamaño de letra:** Usar estándar definidos en las etiquetas h1, h2...etc.
    - **Tipo de fuente:** Helvetia Neue.
  - **Aplicación móvil:**
    - **Tamaño de letra:** 18dp.
    - **Tipo de fuente:** Sans.
- **Colores:** Para el desarrollo de los elementos que conforman nuestro sistema se ha empleado una paleta de colores para asegura de este modo que conseguimos una armonía entre la interfaz de nuestra ampliación móvil y sistema web.

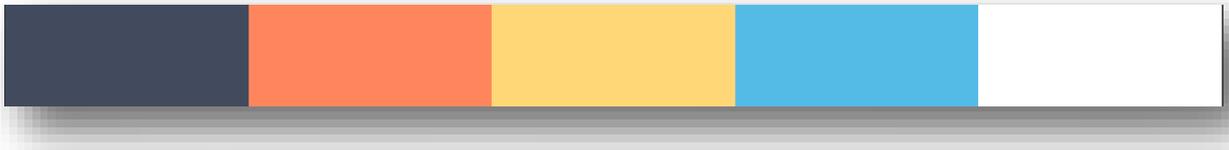


Figura 3.36.- Paleta de colores

- **Iconos:** Los iconos utilizados en el desarrollo de nuestro sistema web son los proporcionados por el framework de diseño Bootstrap [7].



Figura 3.37.- Iconos de Bootstrap

Por otro lado, los iconos utilizados para el desarrollo de nuestra aplicación móvil provienen de Google Desing, en concreto del paquete de iconos Icon Font [8].



Figura 3.38.- Iconos de Google Desing

- **Botones:** Para nuestro sistema web los botones utilizados provienen del framework de desarrollo Bootstrap [9].

## Botones en Bootstrap



Figura 3.39.- Botones de Bootstrap

En cambio, para la aplicación móvil los botones utilizados son los proporcionados por la herramienta de desarrollo utilizada Android Studio.

### 3.4.3.3. Mensajes

En el desarrollo de las interfaces de nuestro sistema debemos tener en cuenta el uso de mensajes ya que es nuestra forma de comunicarnos con el usuario y un mal diseño de los mismos puede suponer el rechazo de nuestro sistema por parte del usuario.

Los mensajes deben de ser educados, concisos, consistentes y constructivos. En nuestro sistema web hemos usado el estilo de mensajes que nos proporciona el framework de diseño Bootstrap [9].



Figura 3.40.- Mensajes Bootstrap

### 3.4.3.4. Storyboard de la aplicación web

El Storyboard nos permite ilustrar de forma fácil como será el comportamiento de nuestro sistema web. A través de un Storyboard se puede representar una secuencia similar a como se comportaría el sistema web con la iteración del usuario. Hay que tener en cuenta que en cada una de las figuras representadas dispondremos de un número que nos indicara donde continuara la iteración del usuario.

En primer lugar, vamos a mostrar la secuencia relacionada con las opciones que el administrador puede realizar en la pantalla principal de nuestro

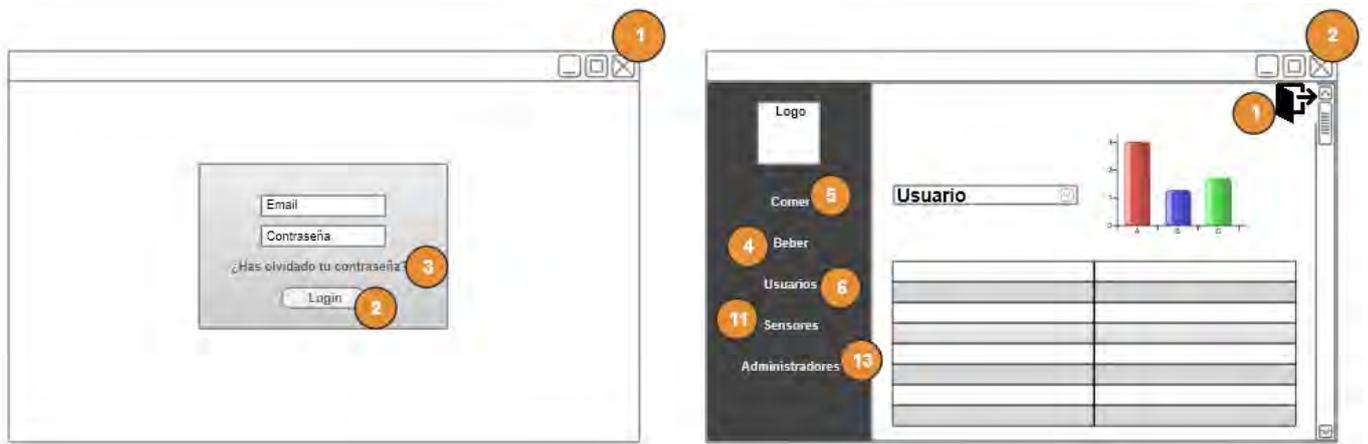


Figura 3.41.-Storyboard sistema web: Inicio sesión 1

sistema web.



Figura 3.42.-Storyboard sistema web: Inicio sesión 2

En segundo lugar, una vez que el administrador se encuentra en su panel de administración, vamos a ver las secuencias más importantes que podría realizar.

- Consultar estado tarea de beber

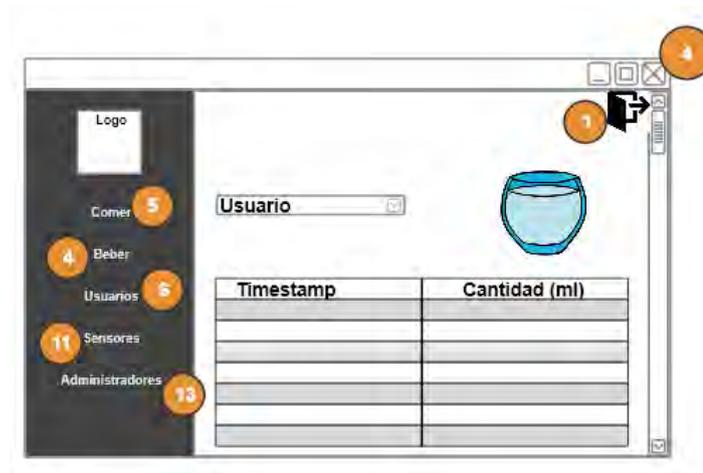


Figura 3.43.-Storyboard sistema web: Estado de la tarea de beber

- Consultar estado tarea de comer



Figura 3.44.-Storyboard sistema web: Estado de la tarea de comer

- Usuarios

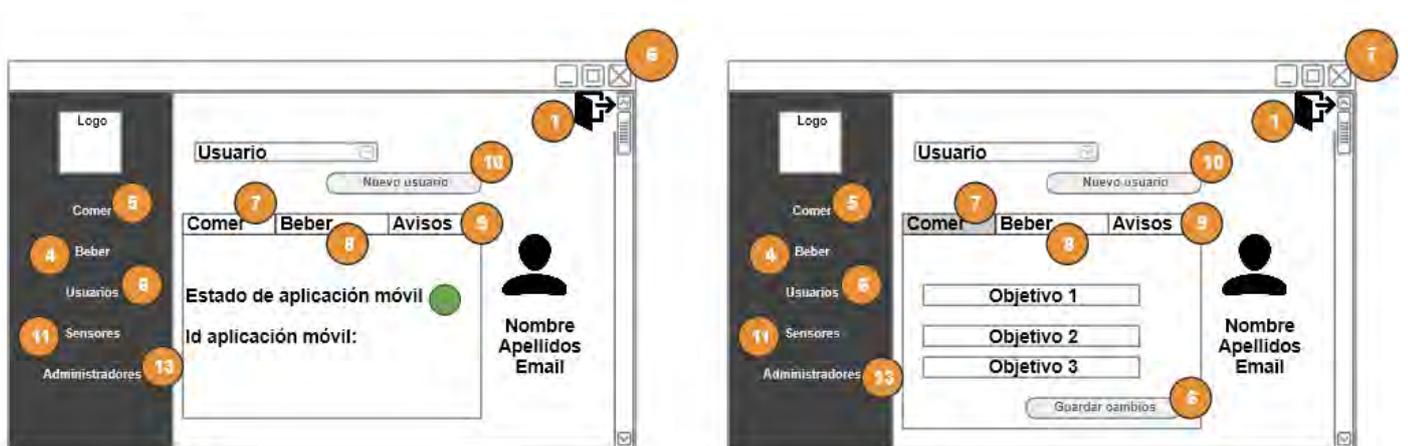


Figura 3.45.-Storyboard sistema web: Usuarios 1



Figura 3.46.-Storyboard sistema web: Usuarios 2

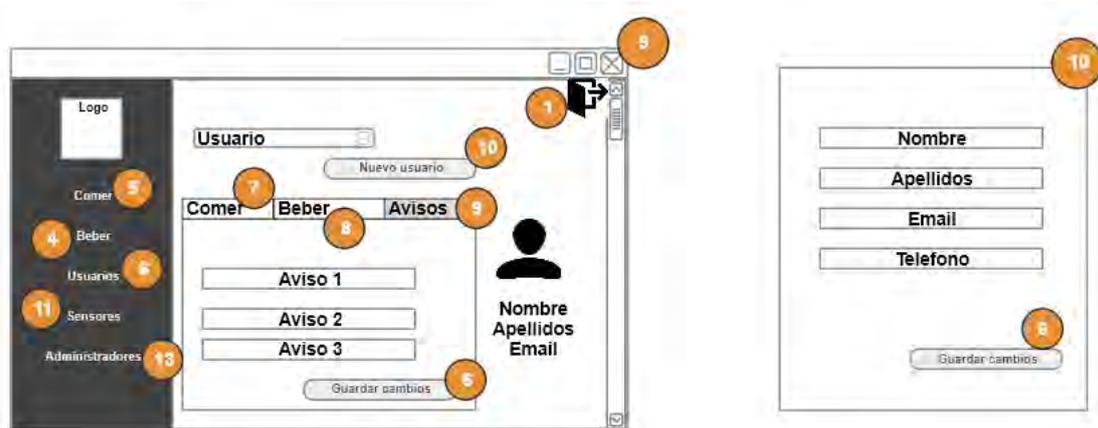


Figura 3.47.-Storyboard sistema web: Usuarios 3

- Dispositivos IoT

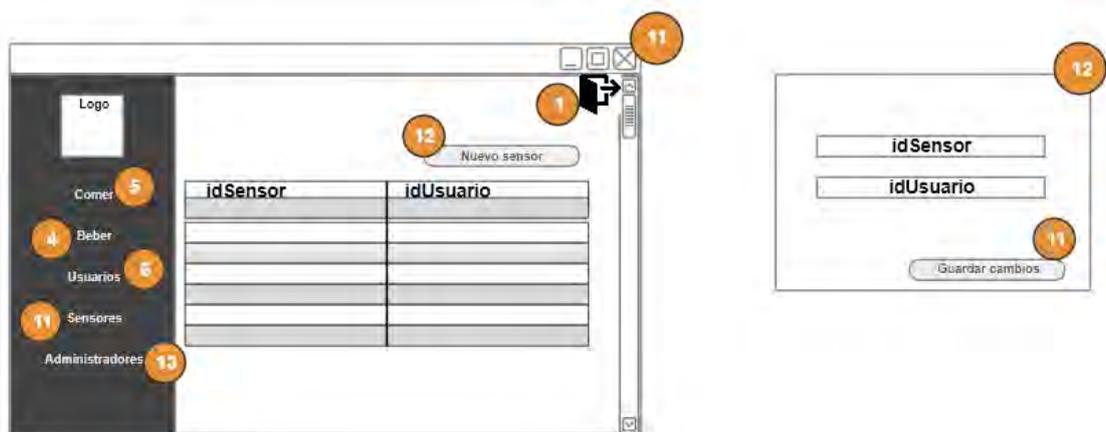


Figura 3.48.-Storyboard sistema web: Dispositivos IoT

- Administradores

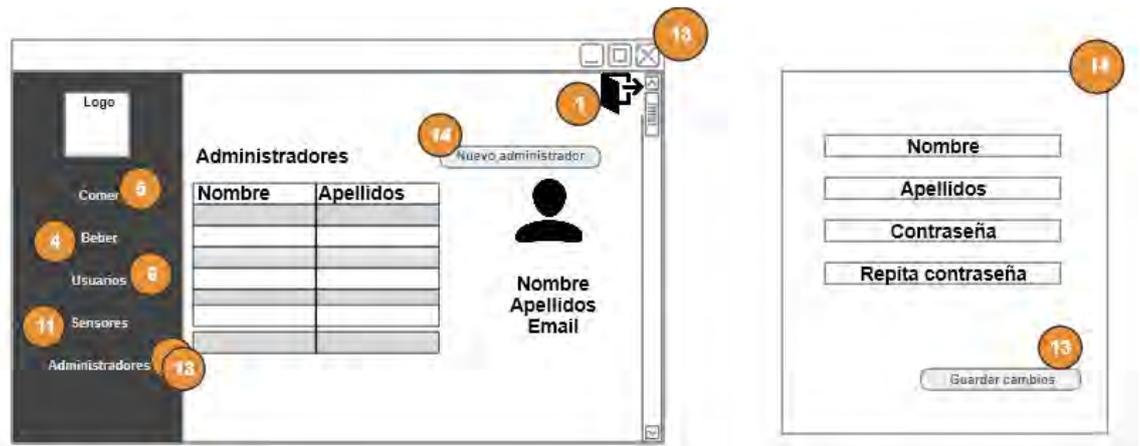


Figura 3.49.-Storyboard sistema web: Administradores

### 3.4.3.5. Storyboard de la aplicación móvil

A continuación, se muestran una serie de figuras que representan las secuencias más importantes que el usuario podrá realizar con nuestra aplicación móvil.

- Identificar al usuario:



Figura 3.50.-Storyboard de la aplicación móvil: Identificar usuario

- Tarea de comer:

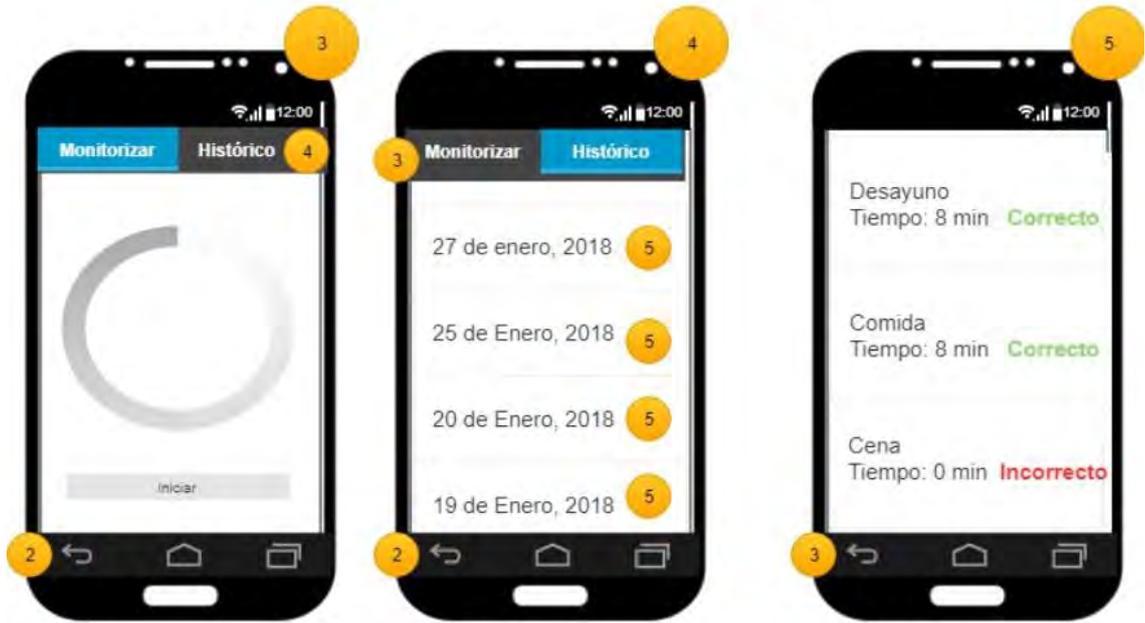


Figura 3.51.-Storyboard de la aplicación móvil: Tarea de comer

- **Tarea de beber:**



Figura 3.52.-Storyboard de la aplicación móvil: Tarea de beber

- **Cierre de sesión:**



Figura 3.53.-Storyboard de la aplicación móvil: Cierre de sesión

## 3.5. Implementación

En la fase de implementación se codifican los diseños y análisis previos del proceso de ingeniería Software empleado en el desarrollo de este proyecto. En esta etapa se traducen todo los modelos que hemos especificado y diseñado anteriormente en código fuente. Cabe destacar que esta etapa debe ser desarrollada de forma correcta ya que un mal desarrollo puede suponer un aumento de costes y tiempo en el desarrollo del proyecto.

### 3.5.1. Lenguajes de programación utilizados

A continuación, se detallan los lenguajes de programación utilizados para el desarrollo de este proyecto.

- **Sistema web:**
  - **PHP:** Es lenguaje de código abierto muy popular, adecuado para el desarrollo web y que puede ser incrustado en HTML [10]. Se trata de un lenguaje interpretado que se ejecuta en la parte servidor

de un sistema y genera de forma dinámica el contenido a petición del cliente.

- **HTML:** Es un lenguaje que pertenece a la familia de lenguajes de marcado y es muy utilizado para la elaboración de páginas web [11]. El estándar HTML lo define la W3C (World Wide Web Consortium) y actualmente se encuentra en su versión HTML5.
- **CSS:** Es un lenguaje de estilo que define la presentación de los documentos HTML [12].
- **Javascript:** Es un lenguaje de programación que a diferencia del lenguaje PHP se utiliza en el lado del cliente, es decir, se ejecuta en el ordenador del usuario no en el servidor, permitiendo crear efectos atractivos y dinámicos en las páginas web [13].
- **Aplicación Móvil:**
  - **Java Android (a diferenciar de Java EE):** Es un lenguaje orientado a objetos cuyo potencial reside en la compilación de código intermedio o bytecode de las aplicaciones desarrolladas haciendo uso de este lenguaje [14].  
Este lenguaje es utilizado por nuestra herramienta de desarrollo Android Studio.
  - **XML:** Se define como un lenguaje de etiquetas en el que cada paquete de información se encuentra delimitado por etiquetas. Android Studio utiliza esta herramienta para definir las vistas o interfaces de una aplicación móvil
- **Sistema de sensores:**
  - **C++:** Aunque en el lenguaje es su lenguaje estándar, no es C++ puro sino que es una adaptación que proviene de avr-libc que provee de una librería de C de alta calidad para usar con GCC en los micros controlores.

### *3.5.2. Herramientas de desarrollo*

Una vez hemos detallado en el apartado anterior los diferentes lenguajes de programación utilizados para el desarrollo de nuestro sistema, procedemos a detallar cada una de las herramientas utilizadas:

- **Sistema web:**

- **CodeIgniter:** Es un framework que permite el desarrollo de aplicaciones en PHP y que utiliza el esquema MVC (modelo-vista-controlador) [15].

Algunos de los puntos más interesantes que nos ofrece este framework son:

- **Compatibilidad:** Destaca por ser compatible con la última versión disponible de PHP, lo que nos permite que pueda ser utilizado en cualquier servidor.
- **Versatilidad:** Puede ser ejecutado en la mayoría de entornos o servidores.
- **Facilidad de instalación:** Solo se requiere subir el framework al servidor y su configuración es muy sencilla.
- **Documentación:** Posee de una amplia documentación que fácil de interpretar y de asimilar.



Figura 3.54.- Logo CodeIgniter

- **Atom:** Se caracteriza por ser un editor de código fuente abierto disponible para múltiples plataformas (Linux, Windows y MacOS). Esta herramienta ha sido utilizada para el desarrollo en PHP de los diferentes componentes que conforma nuestro sistema web.

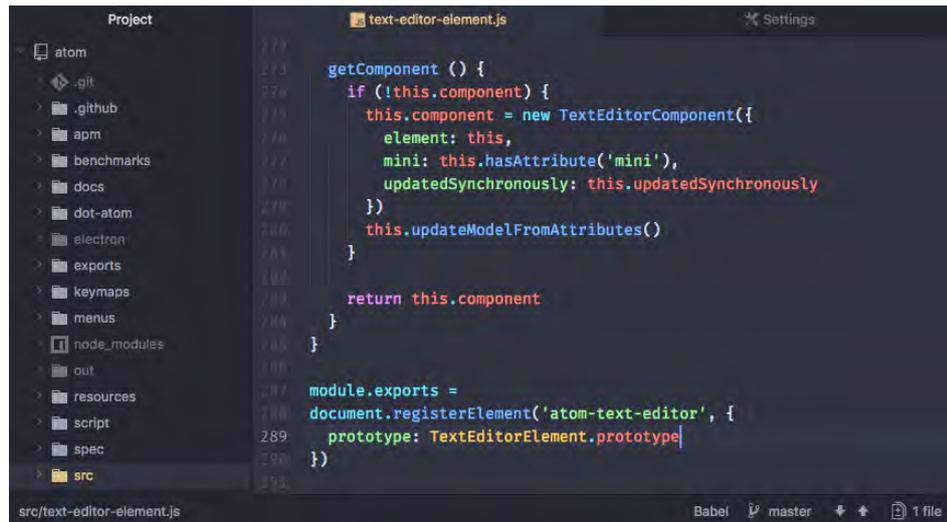


Figura 3.55.- Atom

- **PHPMYAdmin:** Esta herramienta nos permite manejar de forma fácil la administración de MySQL a través de páginas web. Esta escrita en el lenguaje PHP y actualmente tiene un uso muy extendido, se encuentra disponible en múltiples idiomas y posee una licencia GPL [16].

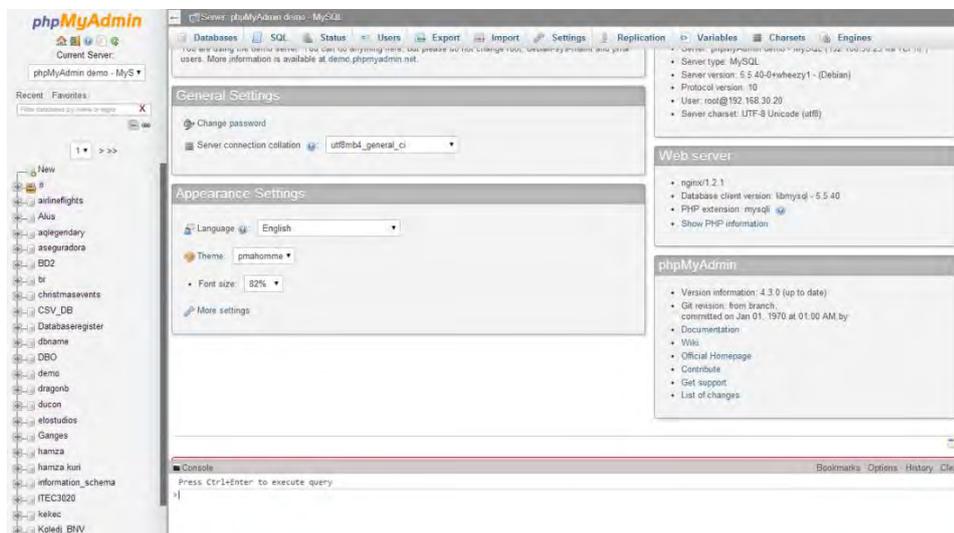


Figura 3.56.- phpMyAdmin

- **Aplicación móvil:**
  - **Android Studio:** Esta herramienta es el entorno de desarrollo integrado (IDE) oficial para el desarrollo de aplicaciones Android y está basado en IntelliJIDEA. Se caracteriza por ser un potente

editor de códigos y por ofrecer múltiples funciones que permiten aumentar la productividad del desarrollador durante la compilación de Apps para el sistema operativo Android.

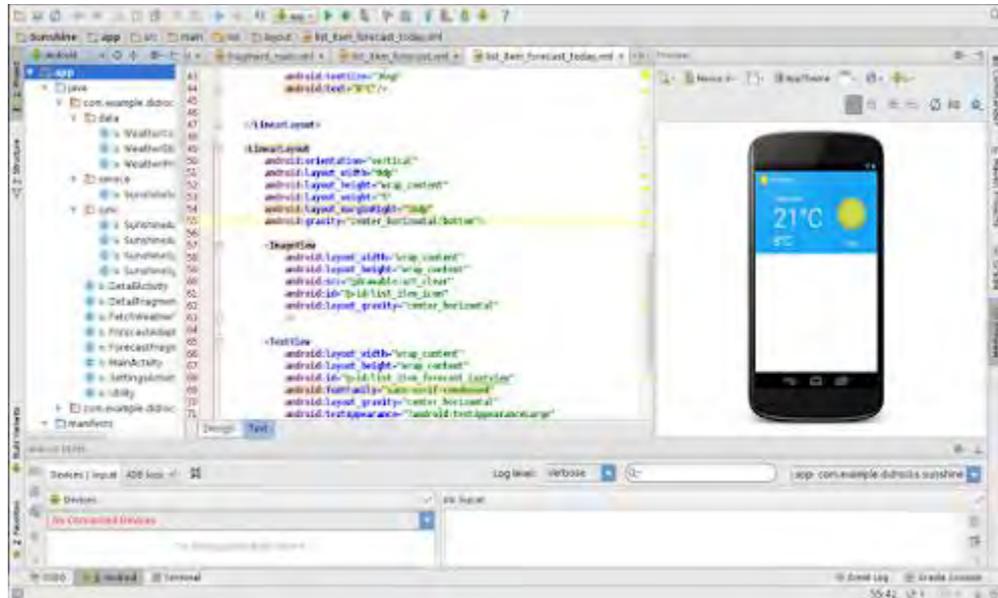
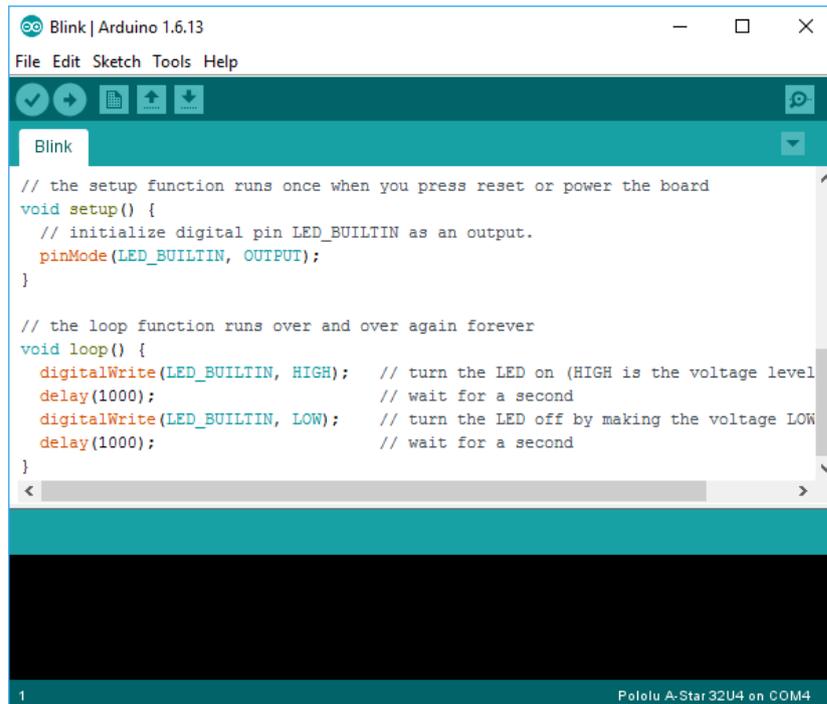


Figura 3.57.- Android Studio

- **Sistema de sensores:**
  - **Arduino 1.8.4:** Constituye un entorno de desarrollo integrado (IDE), consiste en un editor de código, un compilador, un depurador y una interfaz gráfica para facilitar las tareas realizadas en esta herramienta.

The image shows a screenshot of the Arduino IDE interface. The window title is "Blink | Arduino 1.6.13". The menu bar includes "File", "Edit", "Sketch", "Tools", and "Help". Below the menu bar is a toolbar with icons for checkmark, play, document, upload, and download. The main editor area shows the following code:

```
// the setup function runs once when you press reset or power the board
void setup() {
  // initialize digital pin LED_BUILTIN as an output.
  pinMode(LED_BUILTIN, OUTPUT);
}

// the loop function runs over and over again forever
void loop() {
  digitalWrite(LED_BUILTIN, HIGH); // turn the LED on (HIGH is the voltage level)
  delay(1000); // wait for a second
  digitalWrite(LED_BUILTIN, LOW); // turn the LED off by making the voltage LOW
  delay(1000); // wait for a second
}
```

The status bar at the bottom indicates "Pololu A-Star 32U4 on COM4".

Figura 3.58.- Arduino IDE

Para finalizar, quiero destacar que para el desarrollo de los componentes que conforman este proyecto (sistema web, aplicación móvil y sistema de sensores) he utilizado la herramienta de control de versiones SmartGit. El uso de esta herramienta nos ofrece una serie de ventajas a la hora de realizar un desarrollo:

- Permite la posibilidad de restaurar nuestro proyecto a un estado anterior.
- Realiza un registro histórico de cada uno de los cambios realizados en los elementos que conforman el proyecto.

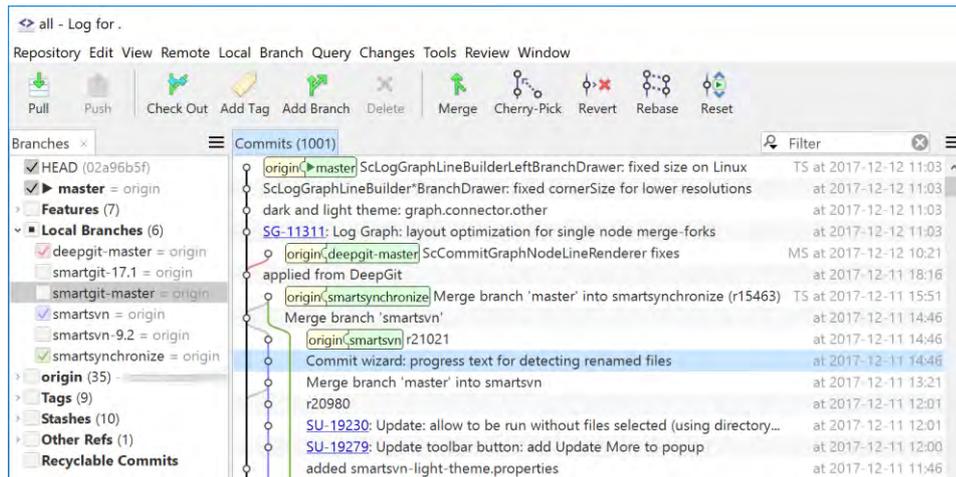


Figura 3.59.- SmartGit

### 3.5.3. Servicios REST implementados

Como se ha mencionado anteriormente, nuestro sistema web ofrece una serie de servicios web que son consumidos por nuestra aplicación móvil o sensores utilizados por los usuarios.

A continuación, se muestra una tabla en la que se describen cada uno de los servicios implementados:

URL	Método	Parámetros	Descripción
<b>/water</b>	GET	Identificador de usuario	Devuelve información sobre la tarea de beber
<b>/waterConfig</b>	GET	Identificador de usuario	Obtiene la configuración de monitorización para las tareas de beber
<b>/eat</b>	GET	Identificador de usuario	Devuelve información sobre la tarea de comer
<b>/eatConfig</b>	GET	Identificador de usuario	Obtiene la configuración de monitorización

			para las tareas de comer
<b>/velocityeat</b>	GET	Identificador del usuario en la app móvil	Obtiene el valor actual de la velocidad con la que el usuario está comiendo
<b>/user</b>	GET	Identificador del usuario en la app móvil	Obtiene los datos personales del usuario
<b>/dayswater</b>	GET	Identificador del usuario en la app móvil	Obtiene el histórico de días en los que el usuario ha realizado las tareas de beber
<b>/dayseat</b>	GET	Identificador del usuario en la app móvil	Obtiene el histórico de días en los que el usuario ha realizado las tareas de comer
<b>/statusEating</b>	GET	Identificador del sensor	Obtiene el estado (iniciada o parada) de la tarea de comer del usuario asociado.
<b>/water</b>	POST	Identificador del sensor y cantidad de líquido ingerida	Almacena un registro procedente de la actividad de beber

<b>/statusEating</b>	POST	Identificador del usuario en la app móvil	Registra que el usuario a iniciado o finalizado la actividad de comer
<b>/registerEating</b>	POST	Identificador del sensor	Almacena un registro procedente de la actividad de comer
<b>/statusapp</b>	POST	Identificador del usuario en la app móvil	Registra el estado de la aplicación móvil

Tabla 3.1.- Servicios REST implementados

Para realizar la implementación de este conjunto de servicios REST en nuestro sistema web bajo el uso del framework CodeIgniter, he tenido que utilizar la librería REST\_Controller. Esta librería es de código abierto y nos ofrece múltiples ventajas, entre las que podemos destacar:

- Permite devolver los datos solicitados en múltiples formatos (XML, HTML, CSV, JSON...etc.).
- Dispone de gestión de seguridad de los servicios REST definido incluyendo posibilidades como la de restringir el uso de un servicio web si el sistema detecta una situación de riesgo
- Realizar un registro de todas las consultas realizadas a los servicios definidos.

### 3.6. Validación

Con esta etapa finalizamos el proceso de ingeniería software empleado en el desarrollo de este proyecto. Las pruebas o validaciones nos permiten comprobar que el software desarrollado cumple de forma correcta con los requisitos especificados, con lo que podemos decir que esta etapa resulta de

gran interés ya que nos permite obtener una valoración de la calidad del software desarrollado y en caso de ser necesario poder realizar mejoras.

Para entender de forma correcta las pruebas realizadas, vamos a diferenciar tres grupos. El primer grupo corresponde a las pruebas realizadas en el sistema web basándonos en los requisitos funcionales descritos en el apartado 2.2.1.1, el segundo grupo a las empleadas en el chequeo de nuestra aplicación móvil también basadas en los requisitos funcionales descritos en el apartado 2.2.2.1 y finalmente, el tercer grupo las pruebas realizadas en el sistema de sensores basándonos en sus requerimientos funcionales especificados en el apartado 2.2.3.1.

- **Pruebas realizadas en el sistema web:**

- **Caso 1: Login correcto de administrador**

<i>Condiciones</i>	El administrador debe estar registrado en el sistema
<b>Acción</b>	El administrador introduce sus credenciales de acceso
<b>Control</b>	El sistema valida los datos introducidos y redirige al administrador a su panel de administración

Tabla 3.2.- Validación del sistema web. Caso 1: Login correcto de administrador

- **Caso 2: Login incorrecto de administrador**

<i>Condiciones</i>	El administrador debe estar registrado en el sistema
<b>Acción</b>	El administrador introduce sus credenciales de acceso
<b>Control</b>	El sistema valida los datos introducidos y muestra un mensaje de error informando al administrador de que los datos introducidos no son correctos

Tabla 3.3.- Validación del sistema web. Caso 2: Login incorrecto de administrador

- **Caso 3: Crear un nuevo administrador de forma correcto**

<i>Condiciones</i>	El administrador no debe estar registrado en el sistema y un administrador registrado debe realizar su registro
<b>Acción</b>	El administrador introduce sus datos en el formulario de registro

<b>Control</b>	El sistema valida los datos introducidos y realiza el registro del nuevo administrador
----------------	--

Tabla 3.4.- Validación del sistema web. Caso 3: Crear un nuevo administrador de forma correcta

- **Caso 4: Crear un nuevo administrador de forma incorrecta**

**Condiciones** El administrador no debe estar registrado en el sistema y un administrador registrado debe realizar su registro

<b>Acción</b>	El administrador introduce sus datos en el formulario de registro
---------------	---

<b>Control</b>	El sistema valida los datos introducidos y muestra al administrador un mensaje de error indicando así que los datos proporcionados no son correctos
----------------	---

Tabla 3.5.- Validación del sistema web. Caso 4: crear un nuevo administrador de forma incorrecta

- **Caso 5: Registro correcto de un nuevo usuario**

**Condiciones** El administrador debe estar identificado en el sistema

<b>Acción</b>	El administrador introduce los datos del usuario en el formulario de registro
---------------	---

<b>Control</b>	El sistema valida los datos introducidos y realiza el registro del nuevo usuario
----------------	--

Tabla 3.6.- Validación del sistema web. Caso 5: Registro correcto de un nuevo usuario

- **Caso 6: Registro incorrecto de un nuevo usuario**

**Condiciones** El administrador debe estar identificado en el sistema

<b>Acción</b>	El administrador introduce los datos del usuario en el formulario de registro
---------------	---

<b>Control</b>	El sistema valida los datos introducidos y muestra un mensaje de error indicando que los datos del usuario introducido no son correctos
----------------	---

Tabla 3.7.- Validación del sistema web. Caso 6: Registro incorrecto de un nuevo usuario

- **Caso 7: Editar un usuario de forma correcta**

<i>Condiciones</i>	El administrador debe estar identificado en el sistema
<b>Acción</b>	El administrador edita los datos o configuración de un usuario
<b>Control</b>	El sistema valida los datos modificados y actualiza los datos almacenados en el sistema sobre el usuario

Tabla 3.8.- Validación del sistema web. Caso 7: Editar un usuario de forma correcta

- **Caso 8: Editar un usuario de forma incorrecta**

<i>Condiciones</i>	El administrador debe estar identificado en el sistema
<b>Acción</b>	El administrador edita los datos o configuración de un usuario
<b>Control</b>	El sistema valida los datos modificados y muestra un mensaje de error indicando que las modificaciones realizadas no son correctas

Tabla 3.9.- Validación del sistema web. Caso 8: Editar un usuario de forma incorrecta

- **Caso 9: Control de la actividad de comer. Consultar datos (correcto)**

<i>Condiciones</i>	El administrador debe estar identificado en el sistema
<b>Acción</b>	El administrador realiza la consulta de los datos registrados en la actividad de comer del usuario
<b>Control</b>	El sistema obtiene los datos y muestra al administrador los datos solicitados

Tabla 3.10.- Validación del sistema web. Caso 9: Control de la actividad de comer. Consultar datos (correcto)

- **Caso 10: Control de la actividad de comer. Consultar datos (incorrecto)**

<i>Condiciones</i>	El administrador debe estar identificado en el sistema
<b>Acción</b>	El administrador realiza la consulta de los datos registrados en la actividad de comer del usuario

**Control**

El sistema obtiene los datos y muestra un mensaje de error indicando que no se ha podido obtener la información requerida
---

Tabla 3.11.- Validación del sistema web. Caso 10: Control de la actividad de comer. Consultar datos (incorrecto)

- **Caso 11: Control de la actividad de beber. Consultar datos (correcto)**

**Condiciones**

El administrador debe estar identificado en el sistema

**Acción**

El administrador realiza la consulta de los datos registrados en la actividad de beber del usuario
--

**Control**

El sistema obtiene los datos y muestra al administrador los datos solicitados
---

Tabla 3.12.- Validación del sistema web. Caso 11: Control de la actividad de beber. Consultar datos (correcto)

- **Caso 12: Control de la actividad de beber. Consultar datos (incorrecto)**

**Condiciones**

El administrador debe estar identificado en el sistema

**Acción**

El administrador realiza la consulta de los datos registrados en la actividad de beber del usuario
--

**Control**

El sistema obtiene los datos y muestra un mensaje de error indicando que no se ha podido obtener la información requerida
---

Tabla 3.13.- Validación del sistema web. Caso 12: Control de la actividad de beber. Consultar datos (incorrecto)

- **Caso 13: Generar de forma correcta identificador de usuario para la aplicación móvil**

**Condiciones**

El administrador debe estar identificado en el sistema

**Acción**

El administrador realiza el registro de un nuevo usuario
--

**Control**

El sistema valida los datos introducidos, registra al usuario y en sus datos muestra el valor del identificador de usuario generado
---

Tabla 3.14.- Validación del sistema web. Caso 13: Generar de forma correcta identificador de usuario para la aplicación móvil

- **Caso 14: Generar de forma incorrecta identificador de usuario para la aplicación móvil. Repetición del identificador en el sistema**

**Condiciones**

El administrador debe estar identificado en el sistema

**Acción**

El administrador realiza el registro de un nuevo usuario
--

**Control**

El sistema valida los datos introducidos, registra al usuario y muestra un mensaje de error indicando que se ha producido un error al generar el identificador del usuario
--

Tabla 3.15.- Validación del sistema web. Caso 14: Generar de forma incorrecta identificador de usuario por la aplicación móvil. Repetición de identificador en el sistema

- **Caso 15: Avisos. Envío de aviso (correcto)**

**Condiciones**

El administrador debe estar identificado en el sistema

**Acción**

El administrador envía una notificación a la aplicación móvil asociada con el usuario
---

**Control**

El sistema realiza el envío de la notificación y muestra un mensaje de envío correcto
---

Tabla 3.16.- Validación del sistema web. Caso 15: Avisos. Envío de aviso (correcto)

- **Caso 16: Avisos. Envío de aviso (incorrecto). Fallo de comunicación**

**Condiciones**

El administrador debe estar identificado en el sistema

**Acción**

El administrador envía una notificación a la aplicación móvil asociada con el usuario
---

**Control**

El sistema realiza el envío de la notificación y muestra un mensaje de error
--

Tabla 3.17.- Validación del sistema web. Caso 16: Avisos. Envío de aviso (incorrecto). Fallo de comunicación

- **Caso 17: Registro de sensor de forma correcta**

*Condiciones* El administrador debe estar identificado en el sistema

<b>Acción</b>	El administrador introduce los datos de un nuevo sensor
---------------	---

<b>Control</b>	El sistema valida los datos introducidos y realiza el registro del nuevo sensor asociado al usuario
----------------	---

Tabla 3.18.- Validación del sistema web. Caso 17: Registro de sensor de forma correcta

- **Caso 18: Registro de sensor de forma incorrecta. Identificador de usuario no correcto**

*Condiciones* El administrador debe estar identificado en el sistema

<b>Acción</b>	El administrador introduce los datos de un nuevo sensor
---------------	---

<b>Control</b>	El sistema valida los datos introducidos y muestra un mensaje de error indicando que no se ha podido realizar el registro del sensor
----------------	--

Tabla 3.19.- Validación del sistema web. Caso 18: Registro de sensor de forma incorrecta. Identificador de usuario no correcto

- **Caso 19: Modificación de datos personales del administrador de forma correcta**

*Condiciones* El administrador debe estar identificado en el sistema

<b>Acción</b>	El administrador realiza modificaciones en sus datos
---------------	--

<b>Control</b>	El sistema valida los datos introducidos y actualiza los datos del administrador
----------------	--

Tabla 3.20.- Validación del sistema web. Caso 19: Modificación de datos personales del administrador de forma correcta.

- **Caso 20: Modificación de datos personales del administrador de forma incorrecta**

*Condiciones* El administrador debe estar identificado en el sistema

<b>Acción</b>	El administrador realiza modificaciones en sus datos
<b>Control</b>	El sistema valida los datos introducidos y muestra un mensaje de error indicando que no se han podido modificar los datos solicitados.

Tabla 3.21.- Validación del sistema web. Caso 20: Modificación de datos personales del administrador de forma incorrecta

- **Pruebas realizadas en la aplicación móvil:**

- **Caso 1: Login correcto del usuario**

<b>Condiciones</b>	El usuario debe estar registrado en el sistema web
<b>Acción</b>	El usuario introduce sus identificador en el formulario de acceso
<b>Control</b>	La aplicación valida el usuario y lo redirige a la vista principal de la aplicación

Tabla 3.22.- Validación de la aplicación móvil. Caso 1: Login correcto del usuario

- **Caso 2: Login incorrecto del usuario**

<b>Condiciones</b>	El usuario debe estar registrado en el sistema web
<b>Acción</b>	El usuario introduce sus identificador en el formulario de acceso
<b>Control</b>	La aplicación valida el usuario y muestra un mensaje de error indicando que el identificador introducido no es correcto

Tabla 3.23.- Validación de la aplicación móvil. Caso 2: Login incorrecto del usuario

- **Caso 3: Control de la actividad de comer (correcto) .Consulta de histórico**

<b>Condiciones</b>	El usuario debe identificado en la aplicación móvil
<b>Acción</b>	El usuario consulta información de los días en los que ha registrado actividad de comer
<b>Control</b>	La aplicación consulta la información solicitada y se la muestra de forma correcta al usuario

Tabla 3.24.- Validación de la aplicación móvil. Caso 3: Control de la actividad de comer (correcto). Consulta de histórico

- **Caso 4: Control de la actividad de comer (incorrecto) .Consulta de histórico (Fallo en la conexión a Internet)**

<i>Condiciones</i>	El usuario debe identificado en la aplicación móvil
<b>Acción</b>	El usuario consulta información de los días en los que ha registrado actividad de comer
<b>Control</b>	La aplicación consulta la información solicitada la aplicación muestra un mensaje de que no es posible en estos momentos mostrar la información solicitada

Tabla 3.25.- Validación de la aplicación móvil. Caso 4: Control de la actividad de comer (incorrecto). Consulta de histórico (Fallo en la conexión de Internet)

- **Caso 5: Control de la actividad de comer (correcto) .Monitorización de la actividad de comer**

<i>Condiciones</i>	El usuario debe identificado en la aplicación móvil
<b>Acción</b>	El usuario inicia la actividad de comer
<b>Control</b>	El aplicación notifica al servidor que el usuario a iniciado la actividad de comer y muestra periódicamente la velocidad registrada por los sensores

Tabla 3.26.- Validación de la aplicación móvil. Caso 5: Control de la actividad de comer (correcto). Monitorización de la actividad de comer

- **Caso 6: Control de la actividad de comer (incorrecto). Monitorización de la actividad de comer (Fallo en la conexión a Internet)**

<i>Condiciones</i>	El usuario debe identificado en la aplicación móvil
<b>Acción</b>	El usuario inicia la actividad de comer
<b>Control</b>	El aplicación muestra un mensaje de error indicando al usuario que en este momento no es posible realizar la acción solicitada

Tabla 3.27.- Validación de la aplicación móvil. Caso 6: Control de la actividad de comer (incorrecto). Monitorización de la actividad de comer (Fallo en la conexión a Internet).

- **Caso 7: Control de la actividad de beber (correcto) .Consulta de histórico**

*Condiciones* El usuario debe identificado en la aplicación móvil

<b>Acción</b>	El usuario consulta información de los días en los que ha registrado actividad de beber
<b>Control</b>	La aplicación consulta la información solicitada y se la muestra de forma correcta al usuario

Tabla 3.28.- Validación de la aplicación móvil. Caso 7: Control de la actividad de beber (correcto). Consulta de historico

- **Caso 8: Control de la actividad de beber (incorrecto) .Consulta de histórico (Fallo en la conexión a Internet)**

*Condiciones* El usuario debe identificado en la aplicación móvil

<b>Acción</b>	El usuario consulta información de los días en los que ha registrado actividad de beber
<b>Control</b>	La aplicación consulta la información solicitada la aplicación muestra un mensaje de que no es posible en estos momentos mostrar la información solicitada

Tabla 3.29.- Validación de la aplicación móvil. Caso 8: Control de la actividad de beber (incorrecto). Consulta de histórico (Fallo en la conexión a Internet)

- **Caso 9: Recibir notificaciones**

*Condiciones* El usuario debe identificado en la aplicación móvil

<b>Acción</b>	El sistema web envía una notificación al usuario
<b>Control</b>	La aplicación móvil recibe la notificación, muestra el mensaje de notificación y emite un sonido de alerta para que el usuario sea notificado.

Tabla 3.30.- Validación de la aplicación móvil. Caso 9: Recibir notificaciones

- **Caso 10: Cierre de sesión**

*Condiciones* El usuario debe identificado en la aplicación móvil

<b>Acción</b>	El usuario cierra la sesión en la aplicación
<b>Control</b>	La aplicación móvil le muestra un mensaje de confirmación y realiza el cierre de la misma

Tabla 3.31.- Validación de la aplicación móvil. Caso 10: Cierre de sesión

- **Pruebas realizadas en el sistema de sensores:**

- **Caso 1: Recogida de datos (correcto)**

*Condiciones* El sensor dispone de datos procesados

<b>Acción</b>	El sensor recoge datos
<b>Control</b>	El sensor almacena los datos y continua su ejecución

Tabla 3.32.- Validación del sistema de sensores. Caso 1: Recogida de datos (correcto)

- **Caso 2: Recogida de datos (incorrecto)**

*Condiciones* El sensor dispone de datos procesados

<b>Acción</b>	El sensor recoge datos
<b>Control</b>	El sensor captura el error y continua su ejecución

Tabla 3.33.- Validación del sistema de sensores. Caso 2: Recogida de datos (incorrecto)

- **Caso 3: Procesamiento de datos (correcto)**

*Condiciones* El sensor dispone de datos procesados

<b>Acción</b>	El sensor intenta procesar los datos recogidos
<b>Control</b>	El sensor almacena los datos calculados y continua su ejecución

Tabla 3.34.- Validación del sistema de sensores. Caso 3: Procesamiento de datos (correcto)

- **Caso 4: Procesamiento de datos (incorrecto).**

*Condiciones* El sensor dispone de datos procesados

<b>Acción</b>	El sensor intenta procesar los datos recogidos
<b>Control</b>	El sensor captura el error y continua su ejecución

Tabla 3.35.- Validación del sistema de sensores. Caso 4: Recogida de datos (incorrecto)

- **Caso 5: Calculo de la información de interés (correcto)**

*Condiciones* El sensor dispone de datos procesados

<b>Acción</b>	El sensor intenta calcular los datos de interés
<b>Control</b>	El sensor almacena los datos calculados y continua su ejecución

Tabla 3.36.- Validación del sistema de sensores. Caso 5: Calculo de información de interés (correcto)

- **Caso 6: Calculo de la información de interés (incorrecto)**

*Condiciones* El sensor dispone de datos procesados

<b>Acción</b>	El sensor intenta calcular los datos de interés
<b>Control</b>	El sensor captura el error y continua su ejecución

Tabla 3.37.- Validación del sistema de sensores. Caso 6: Calculo de la información de interés (incorrecto)

- **Caso 7: Envío de la información al servidor (correcto)**

*Condiciones* El sensor dispone de datos a enviar

<b>Acción</b>	El sensor realiza el intento de envío
<b>Control</b>	El sistema web muestra los datos enviados por el sensor

Tabla 3.38.- Validación del sistema de sensores. Caso 7: Envío de información al servidor (correcto)

- **Caso 8: Envío de la información al servidor (incorrecto). Fallo en la conexión de internet**

*Condiciones* El sensor dispone de datos a enviar

<b>Acción</b>	El sensor realiza el intento de envío
<b>Control</b>	El sensor captura el error y continua su ejecución

Tabla 3.39.-Validación del sistema de sensores. Caso 8: Envío de información al servidor (incorrecto)

### 3.6.1. Resultados

A continuación, procedemos a detallar los resultados obtenidos tras realizar cada una de las pruebas detalladas en el apartado anterior.

- **Resultados obtenidos en las validaciones realizadas en el sistema web:**

Casos	Problemas detectados	Acciones	Resultado
1			Superado
2			Superado
3			Superado
4	No muestra mensaje de error	Capturar mensaje de error	Corregido
5			Superado
6	No muestra mensaje de error	Capturar mensaje de error	Corregido
7			Superado
8			Superado
9			Superado
10			Superado
11			Superado
12			Superado
13			Superado
14			Superado
15	Realiza el envío pero no llega de forma correcta a la aplicación móvil	Solucionar problema encontrado	Corregido
16			Superado
17	No almacena de forma correcta el identificador del usuario	Solucionar problema	Corregido
18			Superado
19			
20			Superado

Tabla 3.40.- Resultados obtenidos en las validaciones del sistema web

- **Resultados obtenidos en las validaciones realizadas en la aplicación móvil:**

Casos	Problemas detectados	Acciones	Resultado
1			Superado
2			Superado
3			Superado
4	La aplicación se cierra de forma inesperada	Capturar mensaje de error	Corregido
5			Superado
6			Superado
7			Superado
8			Superado
9	No se emite sonido de alerta	Cambio en la configuración de las notificaciones	Corregido
10			Superado

Tabla 3.41.- Resultados obtenidos en las validaciones de la aplicación móvil

- **Resultados obtenidos en las validaciones realizadas en el sistema de sensores:**

Casos	Problemas detectados	Acciones	Resultado
1			Superado
2			Superado
3			Superado
4			Superado
5			Superado
6			Superado
7			Superado
8			Superado

Tabla 3.42.- Resultados obtenidos en las validaciones del sistema de sensores

## Capítulo 4

---

### 4. Conclusiones y líneas de trabajo futuras

En este trabajo fin de máster se ha desarrollado un sistema web que permite a un experto en salud poder monitorizar dos hábitos cotidianos que realizan personas que tienen un leve grado de dependencia.

Este sistema web se complementa con una aplicación móvil, la cual ofrece recomendaciones a las personas para conseguir buenos hábitos en dicho contexto, y dos dispositivos IoT para desplegar en los objetos cotidianos asociados a dichos hábitos que permite obtener datos reales de los hábitos realizados por los usuarios y de este modo monitorizarlos.

El dispositivo IoT ha incluido un procesamiento inteligente de datos que ha permitido realizar una motorización continua de las actividades. De esta forma los pacientes pueden tener una respuesta en tiempo real de sus hábitos, así como facilitar el resumen y calidad de los datos a su entorno cercano (familiares y experto en salud).

La propuesta de este trabajo se ha centrado en mejorar dos hábitos cotidianos de personas con un grado de dependencia, en concreto en i) ingesta de alimentos y ii) ingesta de líquidos. Además, el sistema propuesto dota al entorno cercano una herramienta para realizar el seguimiento del paciente a distancia es decir, sin la necesidad de estar presente. También, destacar que a simple vista, pueden parecer muy simples los hábitos que se están monitorizando pero el desempeño de estos de forma incorrecta a largo plazo puede suponer o riesgo para la salud del usuario o agravar los síntomas de otras enfermedades.

Destacamos que en la construcción de este proyecto, ha sido de vital importancia la calidad de los dispositivos IoT desarrollados ya que la fiabilidad de los datos recogidos y el rendimiento de la monitorización de los hábitos de los pacientes vienen determinada por estos dispositivos. Desde mi opinión, considero que en la actualidad resulta muy factible construir este tipo de dispositivos ya que disponemos de una gran variedad de placas

microcontroladores y de múltiples sensores que se le pueden incorporar de forma sencilla. También, hay que destacar que la mayoría del hardware utilizado para la construcción de este tipo de dispositivos en la actualidad posee un precio muy asequible lo que nos permite que esta tecnología pueda ser utilizada por cualquier desarrollador y, lo que es más importante que pueda ser asequible para toda la sociedad sin que exista barrera económica importante para su desarrollo.

Por otro lado, la mayoría de componentes utilizados en la construcción del dispositivo IoT no posee un gran tamaño, ya que es vital que el tamaño y ergonomía de dichos dispositivos son un factor fundamental para conseguir el éxito del sistema desarrollado y su aceptación por parte de los usuarios. No obstante, se hace hincapié en que se ha desarrollado un prototipo no un dispositivo final.

Me gustaría resaltar que este tipo de dispositivos IoT disponen de una amplia variedad de aplicación en este campo y su uso puede aportar una gran cantidad de beneficios en personas con trastornos o enfermedades.

Se ha comprobado con pruebas de evaluación que nuestro sistema funciona de forma correcta y permite conseguir el objetivo para el cual ha sido diseñado. Dicho sistema podría ser extendido para conseguir aumentar su funcionalidad en otros contextos. Es de destacar aquí también que nuestro sistema actualmente es un prototipo con sus limitaciones pero posee un gran potencial de uso y comercialización por explotar.

Por otro lado, sería muy interesante que nuestro sistema aprovechando los sensores desarrollados pudiera realizar la monitorización de otras tareas de gran importancia, como puede ser el caso de saber si el usuario se mueve en deshoras, se lava los dientes o coge un determinado elemento de la vivienda. Además, nuestros sensores poseen conexión Bluetooth lo que nos podría permitir conocer la posición de usuario en la vivienda haciendo uso de unas determinadas balizas como son las Beacons y esto, nos podría permitir saber si la persona visita diferentes estancias de la vivienda de interés para su salud como puede ser el cuarto de baño.

Así mismo, se podría aumentar la funcionalidad de nuestra aplicación móvil para recoger datos de los sensores del dispositivo móvil en que se ejecuta dicha aplicación y obtener así, datos de interés para poder realizar la monitorización de otras actividades, como puede ser los pasos que el usuario realiza en un día.

Otro aspecto a ampliar sería la interfaz de nuestra aplicación móvil ya que su visualización en dispositivos con un tamaño superior a 10 pulgadas podría mejorarse.

Para finalizar, como conclusión personal, me gustaría poder expresar mi gran satisfacción de haber podido desarrollar este trabajo fin de máster, ya que ha supuesto un reto personal y una posibilidad de poder desarrollar un sistema que ayude a las personas a mejorar su calidad de vida, lo que supone una gran satisfacción para mí. La realización de este trabajo me ha permitido conocer nuevas tecnologías que desde mi punto de vista dispondrá en los años próximos de una gran implantación.

## Referencias

---

- [1] “Weiser, M.: The computer for the 21st century. Sci. Am. 265(3), 94–104 (1991)”
- [2] “Javier Medina, Macarena Espinilla, Daniel Zafra, Luis Martinez and Christopher Nugent: Fuzzy Fog Computing: A Linguistic Approach for Knowledge Inference in Wearable Devices, (2017)”
- [3] “Wemos”.<https://wiki.wemos.cc/products:lolin32:lolin32>
- [4] “Arduino”.<https://playground.arduino.cc/Main/MPU-6050>
- [5] “[http://www.naylampmechatronics.com/blog/45\\_Tutorial-MPU6050-Aceler%C3%B3metro-y-Giroscopio.html](http://www.naylampmechatronics.com/blog/45_Tutorial-MPU6050-Aceler%C3%B3metro-y-Giroscopio.html)”
- [6] “robologs”. <https://robologs.net/2014/10/15/tutorial-de-arduino-y-mpu-6050/>
- [7] “Bootstrap”. <http://getbootstrap.com/css/>
- [8] “Google Desing”. <https://design.google.com/>
- [9] “w3schools”. <http://www.w3schools.com/bootstrap/>
- [10] “php”. <http://php.net/manual/es/intro-what-is.php>
- [11] “Acerca de html”. <http://www.acercadehtml.com/manual-html/que-es-html.html>
- [12] “w3c”. <http://www.w3c.es/Divulgacion/GuiasBreves/HojasEstilo>
- [13] “Librosweb”. [http://librosweb.es/libro/javascript/capitulo\\_1.html](http://librosweb.es/libro/javascript/capitulo_1.html)
- [14] Jon Byous, Java technology: “The early years. Sun Developer Network”, sin fecha [ca. 1998]. Recuperado 21 de abril de 2005.
- [15] “Asociación de desarrolladores web de España”.  
<http://www.adwe.es/codigo/codeigniter-framework-php-desarrollo-aplicaciones-web>
- [16] “phpmyadmin”. <https://www.phpmyadmin.net/>

## Anexos

---

### A. Contenido del CD-ROM

En este apartado, se detalla el contenido del disco adjunto en la documentación de esta memoria de trabajo fin de máster.

- **Aplicaciones finales:**
  - **Aplicación web:** Carpeta con el contenido de nuestro sistema web.
  - **Aplicación móvil:** Fichero .apk para permitir la instalación de nuestra aplicación móvil en un dispositivo Android.
  - **Sensores:** Contiene los ficheros .ino necesarios para poder ser instalados en un microcontrolador que posea el sensor requerido para su funcionamiento.
- **Base de datos:** En esta carpeta se almacenan múltiples ficheros con extensión .SQL. Estos archivos contienen una copia actual de toda la base de datos del sistema.
- **Código Fuente:** En esta carpeta podemos encontrar el código perteneciente a cada una de las aplicaciones desarrolladas en este proyecto. El código de nuestro sistema web se puede abrir con cualquier editor de texto, para abrir el código perteneciente a nuestra aplicación móvil debemos usar la IDE Android Studio y finalmente, para abrir el código de los sensores debemos usar IDE de desarrollo de Arduino.
- **Memoria:** Fichero pdf que contiene la documentación desarrollada en este trabajo fin de máster.
- **Demostración de funcionamiento:** Contiene un vídeo demostrativo del funcionamiento de nuestro sistema.

### B. Manual de instalación

En el siguiente apartado, vamos a describir los pasos necesarios para conseguir desplegar cada una de las aplicaciones que constituyen el sistema desarrollado en este proyecto.

Para entender de forma correcta este manual, vamos a diferenciar entre los pasos a seguir para la instalación de nuestro sistema web, los pasos necesarios para nuestra aplicación móvil y finalmente, los pasos necesarios para los sensores.

## B.1. Sistema web

Antes de realizar la instalación de nuestro sistema web, debemos tener en cuenta que es necesario disponer de un servidor con soporte para PHP, MySQL y preferiblemente que disponga de una distribución Linux.

En primer lugar, debemos realizar la creación de la base de datos necesaria para nuestro sistema web. Para ello, podemos usar la herramienta phpMyAdmin que nos permite realizar el proceso de una forma fácil e intuitiva. Debemos iniciar nuestro gesto de base de datos y pulsar en el botón nueva de la barra lateral de nuestro panel de administración (véase en la siguiente figura).



Figura manuales 1.- Creación de la base de datos en phpMyAdmin

Seguidamente, crearemos una base de datos con el nombre “smartneur” y un usuario, diferente al root, con privilegios para acceder a nuestra base de datos. Para ello, debemos pulsar en nuestra base de datos y después, en el menú superior de la pantalla de administración de nuestra base de datos pulsar en la opción *privilegios* y seguidamente en la opción *Agregar usuario*. El usuario creado deberá disponer de todos los privilegios.

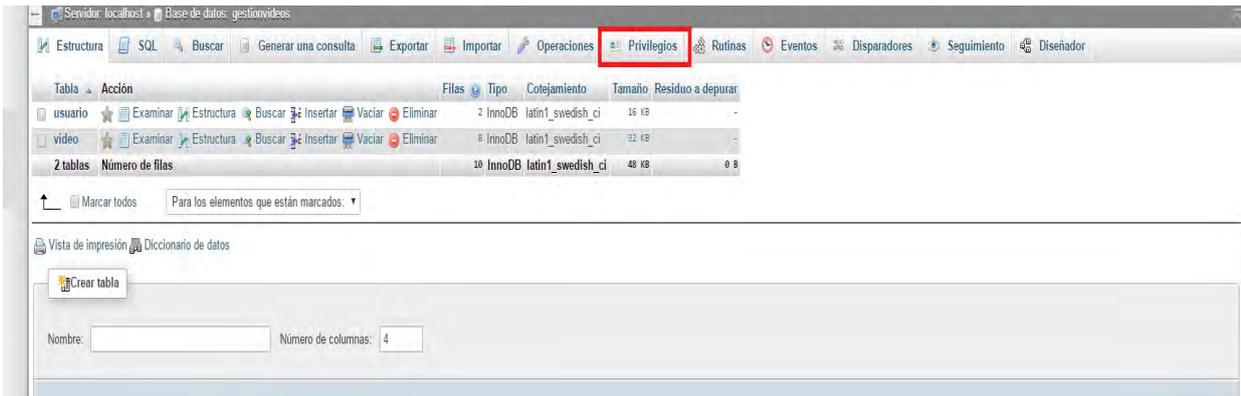


Figura manuales 2.-Agregar usuario a la base de datos

Una vez hayamos creado la base de datos, debemos importar los archivos con extensión .sql que se aporta en el CD-ROM. Al importar estos archivos, nos creara las tablas necesarias para la utilización de nuestro sistema web.

Para importar un archivo, debemos pulsar en el botón *importar* del panel de administración de nuestro sistema web y después, seleccionar el archivo .sql que se desea importar.

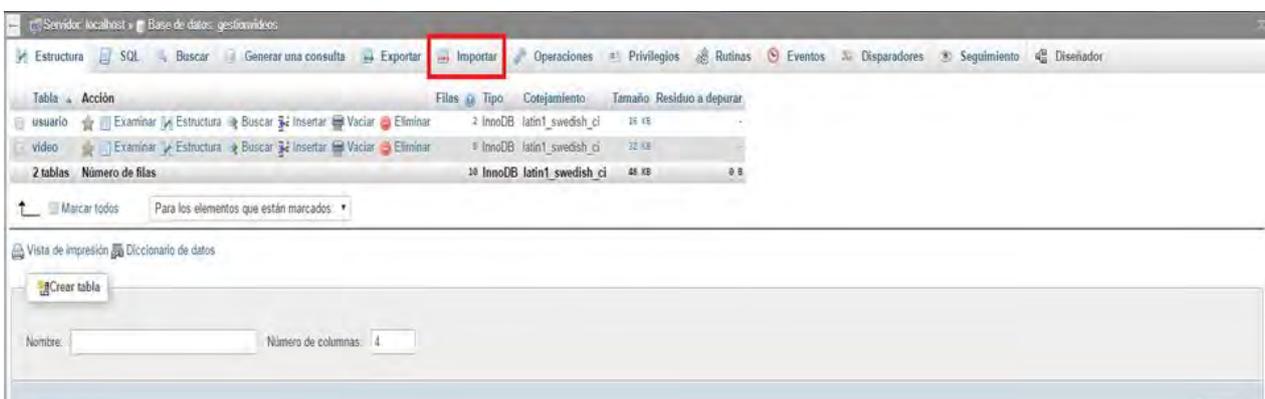


Figura manuales 3.- Importar fichero .sql

El segundo lugar, debemos conectarnos por FTP a nuestro servidor y copiar el contenido de la carpeta de código fuente de nuestro sistema web en la carpeta www de nuestro servidor Apache. Para ello podemos hacer uso de algún cliente FTP como puede ser FileZilla.

Una vez se hay completado el proceso de subida de los archivos de nuestro sistema web, debemos realizar los siguientes cambios:

1. Debemos cambiar la configuración de nuestra aplicación web. Para ello, debemos modificar el archivo “config.php” localizado en la siguiente dirección “SmartNeur/application/config/config.php”, en el cual realizaremos el siguiente cambio:
  - a. `$config['base_url'] = 'http://serezade.ujaen.es:8013';`
2. Debemos configurar la conexión a nuestra base de datos. Para realizar esta acción, nos dirigimos al archivo “database.php” localizado en la dirección “SmartNeur/application/config/database.php” y realizaremos los siguientes cambios:
  - a. `'username'=>'nombre del usuario que hayamos creado'`
  - b. `'password'=>'la contraseña del usuario'`
3. Debemos cambiar la url de los servicios consumidos por los Javascript de nuestra aplicación. Para ello, nos dirigiremos a la capeta “js” del código fuente de nuestro proyecto y revisaremos las URLs definidas en cada uno de los archivos .js y en caso de ser necesarios cambiar dichas URls por nuestra URL base (en nuestro caso `http://serezade.ujaen.es:8013`).

Finalmente, si hemos realizado de forma correcta todos los pasos detallados en este manual, podremos acceder a nuestro sistema web.

## B.2. Aplicación móvil

Para instalar nuestra aplicación móvil, solo es necesario ejecutar el archivo .apk en un dispositivo móvil o Tablet y el proceso de instalación de nuestra aplicación se iniciara de forma automática.

Por otro lado, se puede realizar la instalación de nuestra aplicación a través del IDE Android Studio, haciendo uso del código fuente. Para ello

debemos conectar por USB nuestro dispositivo móvil o Tablet a nuestro ordenador y activar el modo depuración de nuestro dispositivo. Seguidamente, abriremos con Android Studio el código fuente de nuestra aplicación móvil y para finalizar, solo tendremos que pulsar en el botón *ejecutar* de nuestro proyecto en Android Studio y la aplicación comenzara a instalarse en nuestro dispositivo móvil o Tablet.

### B.3. Sensores

Para realizar la instalación de los sensores debemos instalarnos el IDE de Arduino en nuestro ordenador y seguidamente, nos dirigiremos a la carpeta de código fuente de nuestro CD y abriremos con el IDE de Arduino cada una de las clases .ino desarrolladas para nuestros sensores.

Debemos tener en cuenta que para el correcto funcionamiento de nuestros sensores es necesario disponer de una red Wifi accesible. Podemos crear esta red de tres formas distintas:

1. Creando una red Wifi a través de nuestro dispositivo móvil. En este caso habrá que crear dicha red con las siguientes propiedades:
  - **SSID:** WifiSensores
  - **Contraseña:** Sensores2018
2. Creando una red Wifi en el Router de la vivienda. En este caso la red creada debe disponer de las mismas propiedades que en el caso anterior.
3. Modificando la red por defecto del sensor para que se conecte a una red ya existente. Para realizar esta acción debemos abrir el fichero .ino de nuestro sensor a través de la IDE de desarrollo de Arduino y modificar los datos de la red Wifi a la que se va a conectar. En la siguiente imagen se puede visualizar los datos que se deben modificar:

```
//Datos de la red wifi
const char* ssid = "MIWIFI_2G_qDqh";
const char* password = "hsmk7raexdcb";
```

Figura manuales 4.- Modificación de Wifi

Por otro lado, cada sensor que este registrado en nuestro sistema web, debe poseer un identificador único. Es por ello, que debemos modificar el identificador del sensor en su propio código por su identificador impreso (pegativa con código en el sensor). En la siguiente imagen se puede observar de forma más sencilla el dato que se debe modificar:

```
String stringOne = "{\"idSensor\":\"AEX5621KL\"}";
```

Figura manuales 5.-Modificación del identificador del sensor

Para finalizar, conectaremos por USB nuestra placa microcontroladora a nuestro ordenador y pulsaremos la opción *subir* del IDE de Arduino para que de este modo comience la instalación.



Figura manuales 6.- Subir código Arduino IDE

## C. Manual de usuario del sistema web

### C.1. Acceso al sistema

Para poder acceder a nuestro sistema web, el usuario (administrador) deberá rellenar de forma correcta el formulario de acceso (véase en la Figura de manuales 7) que se muestra en la pantalla principal de nuestro sistema web.

The image shows a login form titled 'ACCESO' in a light blue header. Below the header, there are two input fields: 'Username' and 'Contraseña'. A link for '¿Olvidaste de la contraseña?' is positioned below the password field. At the bottom, there is a large blue button labeled 'Iniciar sesión' with a lock icon.

Figura manuales 7.- Formulario de acceso al sistema web

Si los datos proporcionados son correctos, el sistema de forma automática lo redirigirá a su panel de administración (Figura de manuales 8).

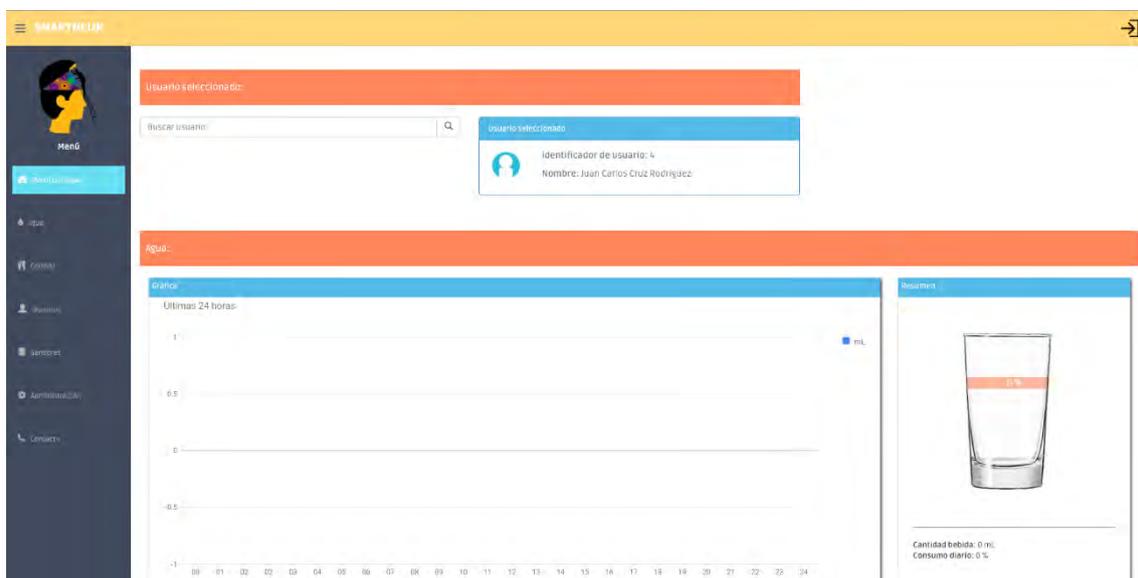


Figura manuales 8.- Panel de administración del sistema web

## C.2. Recuperación de las credenciales de acceso

En el caso de que no recordemos las credenciales de acceso a nuestro sistema web. El sistema le proporciona al usuario la posibilidad de recuperar su contraseña, para ello el usuario deberá pulsar en la opción *¿Olvidaste la contraseña?* del formulario de acceso y le mostrará el siguiente formulario (Figura manuales 9) donde el usuario deberá introducir su cuenta de correo con la que este registrado en nuestro sistema.

The image shows a web form for password recovery. At the top, there is a light blue header with the text "¿Olvidaste de la contraseña?". Below the header, the instruction "Introduce tu email para recuperar la contraseña." is displayed. A text input field with the placeholder "Email" is provided for the user to enter their email address. At the bottom right of the form, there are two buttons: a white "Cancelar" button and a blue "Aceptar" button.

Figura manuales 9.- Formulario para recordar credenciales

Si los datos proporcionados por el usuario son correctos, el sistema enviará de forma automática a la cuenta de correo del usuario un mensaje con los pasos para reestablecer la contraseña.

### C.3. Registro de un nuevo administrador (cuidador)

Para poder realizar el registro de un nuevo administrador o cuidador en el sistema, previamente otro administrador debe realizar dicho registro. Para ello debe pulsar en la opción *Administración* del panel principal de nuestro sistema web y seguidamente, pulsar en la opción *nuevo administrador* y rellenar de forma correcta con los datos del nuevo administrador el formulario de registro proporcionado por el sistema (véase en la Figura manuales 10).



Figura manuales 10.- Formulario de registro de un nuevo administrador

Si los datos del nuevo administrador han sido introducidos de forma correcta, el sistema debe poder permitir el acceso al sistema del nuevo administrador.

#### C.4. Registro de un nuevo usuario a monitorizar

Para poder monitorizar las tareas de un usuario debemos realizar su registro en nuestro sistema web. Para ello el administrador o cuidador debe dirigirse a la pestaña *usuarios* dentro del panel de administración (véase en la figura de manuales 8) y seguidamente, pulsar en la opción *nuevo usuario* (véase en la siguiente figura).

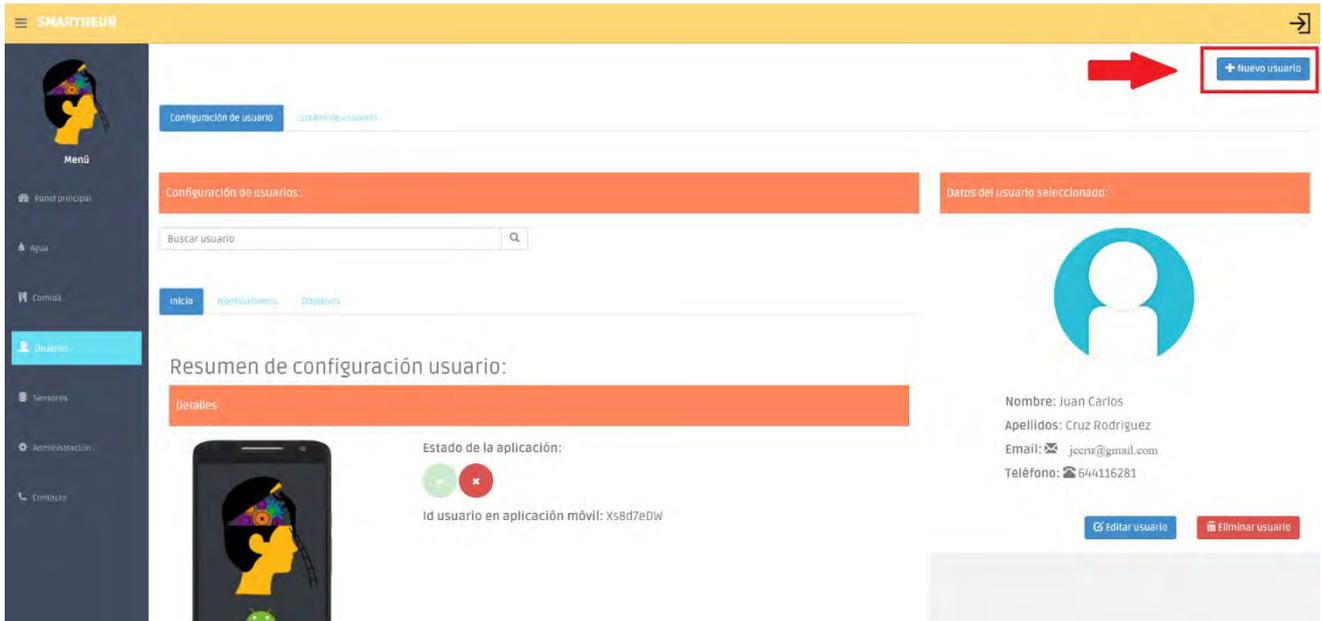


Figura manuales 11.- Registro de nuevo usuario

Finalmente, el cuidador debe rellenar de forma correcta el formulario de registro de nuevo usuario que nuestro sistema web le proporciona.

The 'Nuevo usuario' form is displayed with a light blue header. Below the header, there is a instruction: 'Rellene de forma correcta los campos requeridos en el siguiente formulario.' The form contains four input fields: 'Nombre', 'Apellidos', 'Teléfono', and 'Email'. At the bottom right, there are two buttons: 'Crear usuario' (blue) and 'Cerrar' (red).

Figura manuales 12.- Formulario de registro de un nuevo usuario

### *C.4.1. Configuración de objetivos del usuario en las tareas monitorizadas*

Una vez hemos realizado el registro de un nuevo usuario, debemos realizar una configuración inicial de las preferencias de monitorización en las tareas monitorizadas de este usuario. Para ello, debemos pulsar en la pestaña *objetivos* (véase en la figura de manuales 11) y rellenar de forma correcta los campos requeridos para dicha configuración.

Inicio Notificaciones **Objetivos**

## Configuración de objetivos:

### Agua

**Objetivos**  
Establece la cantidad de líquido diaria recomendada para el usuario.

Cantidad en mililitros

**Guardar**

### Comer

**Objetivos**  
Establece la franja horaria en la que el usuario debe comer.

**Desayuno**  
Hora Inicio:  Hora Fin:

**Comida**  
Hora Inicio:  Hora Fin:

**Cena**  
Hora Inicio:  Hora Fin:

**Guardar**

Figura manuales 13.- Configuración de objetivos

Finalmente, también podremos configurar los avisos que el usuario recibirá. Para ello debemos pulsar en la pestaña avisos y rellenar de forma correcta los campos solicitados.

## Configura las notificaciones:

### Agua

**Notificaciones**  
Envía una notificación al teléfono móvil o tablet suscrito para recordar al usuario que debe de tomar una poco de líquido.

**Opciones**

Recibir Notificaciones:

Activado  Desactivado

Aviso en minutos

Intervalo horario

Hora Inicio:  Hora Fin:

**Guardar**

### Comida

**Notificaciones**  
Envía una notificación al teléfono móvil o tablet suscrito para recordar si no ha comido en un intervalo horario.

**Opciones**

Recibir Notificaciones:

Activado  Desactivado

Aviso en minutos



Figura manuales 14.- Configuración de avisos

### C.5. Control de las tareas de beber

El cuidador o administrador puede comprobar el estado de las tareas de beber de los usuarios monitorizados. Para realizar esta acción, el administrador debe pulsar en la opción *agua* del panel de administración del sistema web (véase en la figura de manuales 8) y seguidamente, buscar al usuario del cual quiere consultar la información referente a sus tareas de beber. Finalmente, el sistema le muestra al usuario la información referente al usuario solicitado.

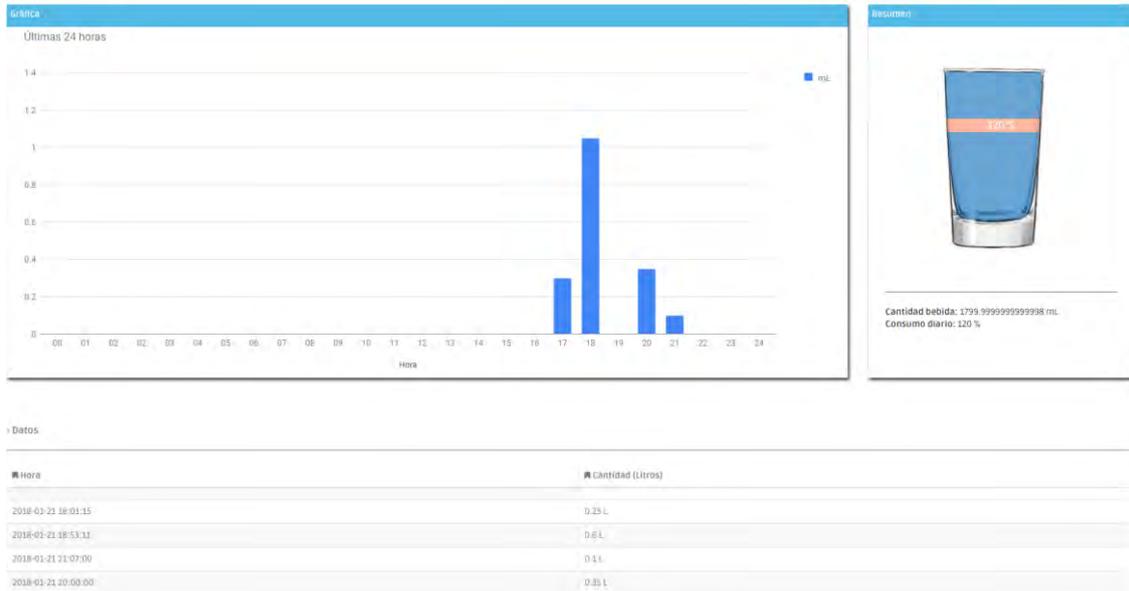


Figura manuales 15.- Información sobre las tareas de beber

## C.6. Control de las tareas de comer

De una forma muy similar al caso anterior, para consultar el estado de las tareas de comer de un usuario registrado en el sistema, el cuidador o administrador deberá pulsar en la opción *comida* del panel de administración del sistema web (véase en la figura de manuales 8) y seguidamente, seleccionar al usuario del cual quiere realizar la consulta. Finalmente, el sistema le mostrara la información solicitada.

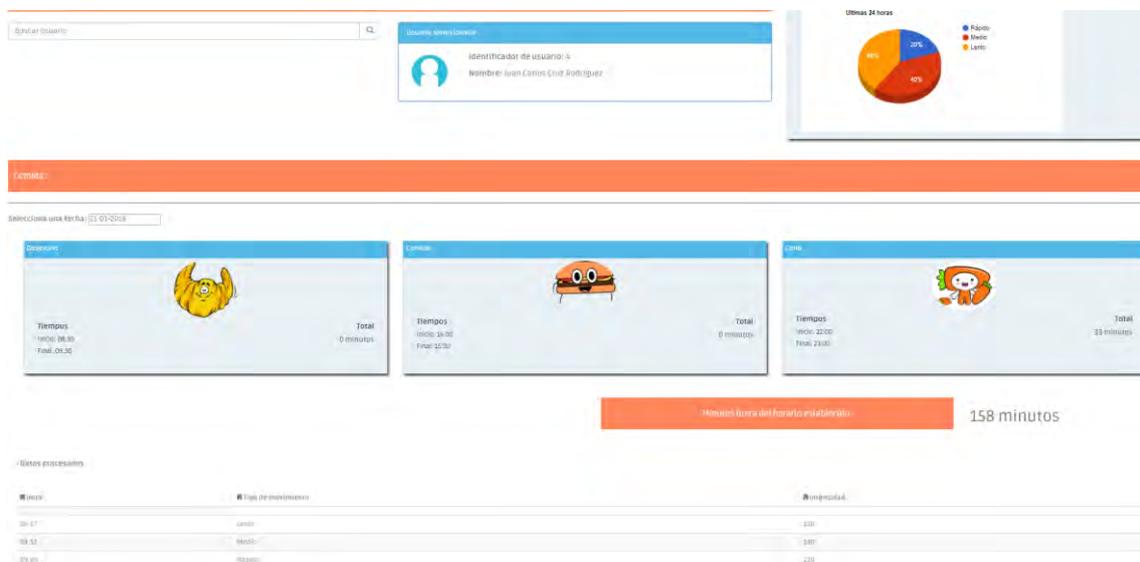


Figura manuales 16.- Información sobre las tareas de comer

## C.7. Registro de un nuevo sensor

El cuidador podrá registrar un nuevo sensor utilizado por usuario en sus tareas monitorizadas. Para ello el administrador se dirigirá a la pestaña *sensores* del panel de administración del sistema web (véase en la figura de manuales 8) y pulsará en la opción *nuevo sensor*. Finalmente, se debe rellenar de forma correcta los campos del formulario que el sistema nos muestra para el registro de un nuevo sensor.



Figura manuales 17.- Pantalla sensores

Figura manuales 18.- Formulario de registro de un nuevo sensor

## D. Manual de usuario de la aplicación móvil

### D.1. Acceso a la aplicación móvil

Para poder acceder a pantalla principal de nuestra aplicación móvil, el usuario deberá introducir su identificador de sesión (proporcionado por el administrador o cuidador) en la pantalla de inicio de sesión de nuestra aplicación (véase en la siguiente figura).

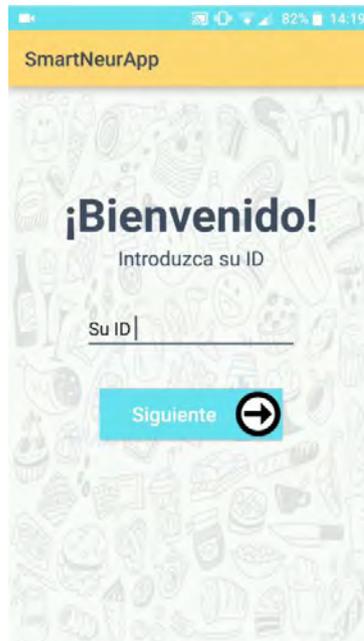


Figura manuales 19.- Identificación en la aplicación móvil

Finalmente, si el usuario ha introducido de forma correcta su identificador, la aplicación móvil le redirigirá de forma automática a la pantalla principal de nuestra aplicación.

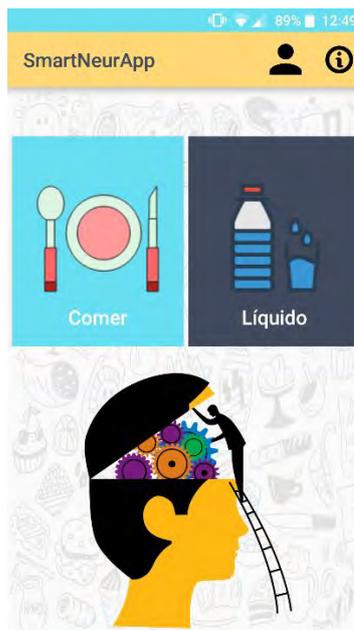


Figura manuales 20.- Pantalla principal de la aplicación móvil

## D.2. Control de la actividad de comer

El usuario podrá realizar dos operaciones para poder llevar el control en sus tareas de comer: control de la velocidad con la que comer en tiempo real y consulta del histórico de las tareas realizadas en días anteriores.

A continuación, se detalla cada una de las dos opciones indicadas anteriormente.

### D.2.1. Control de la velocidad de comer

Para realizar un control de la velocidad con la que el usuario realizar su actividad de comer, el usuario deberá dirigirse a la sección *comer* de nuestra aplicación móvil y seguidamente en la pestaña monitorizar pulsar en el *botón iniciar* (véase en la figura de manuales 21) para que de este modo los sensores empiecen a recoger información.

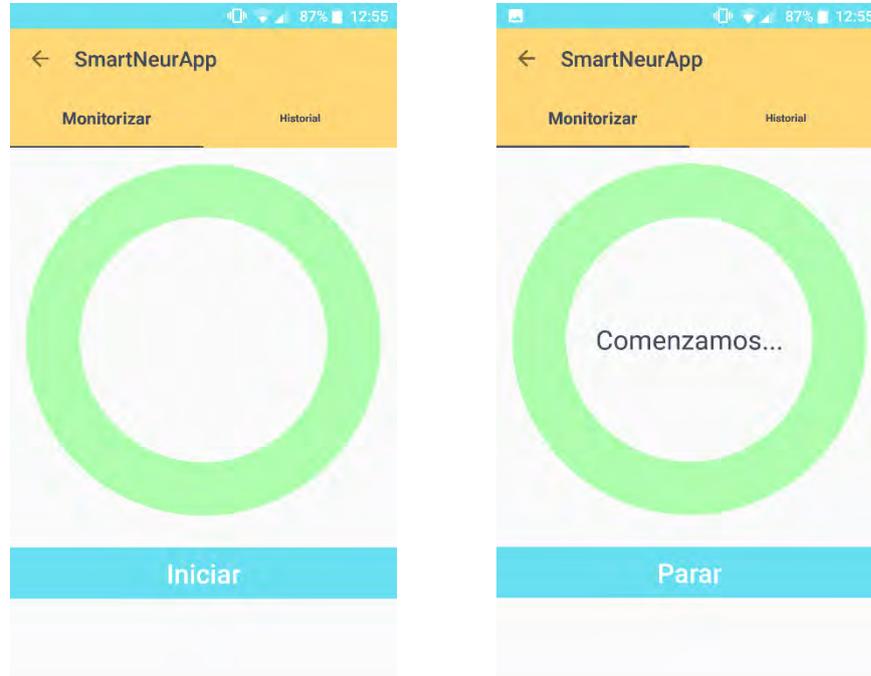


Figura manuales 21.- Monitorización de las tareas de comer

Mientras el usuario realiza la acción de comer la aplicación móvil le ira informando de la velocidad con la que está realizando dicha tarea.

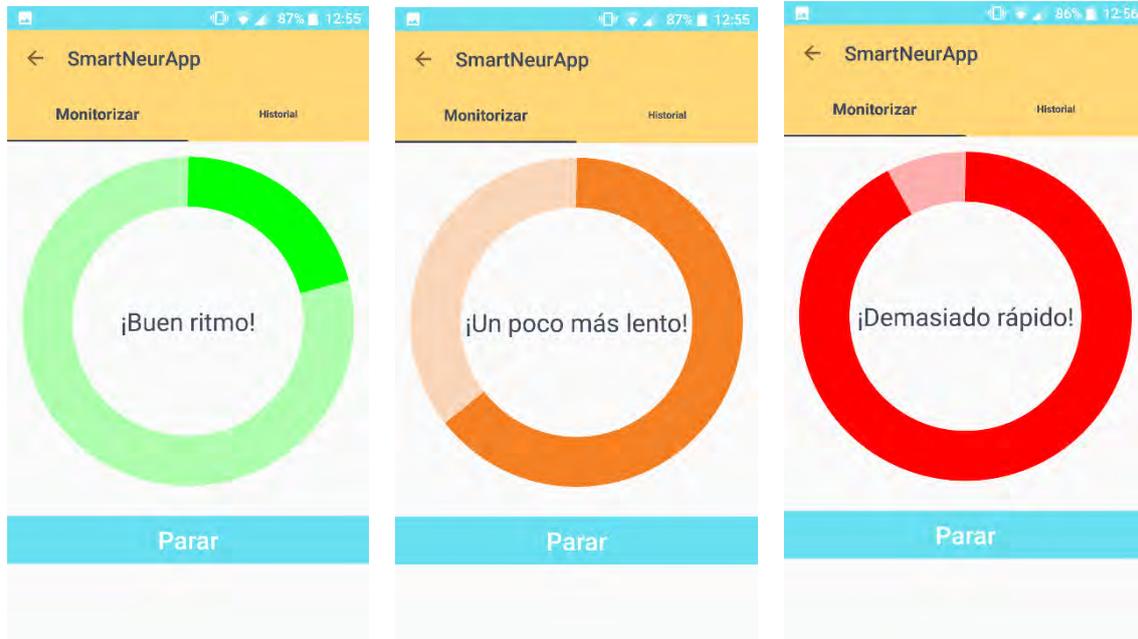


Figura manuales 22.- Información sobre la velocidad de comer

Finalmente, el usuario pulsara en el botón parar para indicar de este modo que ha terminado su tarea de comer.

### *D.2.2. Consultar histórico*

Para consultar el histórico de tareas de comer realizadas en días anteriores, el usuario deberá dirigirse a la pestaña *Historial* de la sección *Comida* de nuestra aplicación móvil y seguidamente, nuestra aplicación muestra la información de la que dispone.



Figura manuales 23.- Histórico de las tareas de comer

Finalmente, para obtener el detalle de un día del historial, el usuario solo debe pulsar en el día elegido y nuestra aplicación le mostrara la información recogida sobre el día seleccionado.



Figura manuales 24.-Detalle de las tareas de comer

## D.3. Control de la actividad de beber

De la misma forma que en el caso anterior, en el control de la actividad de beber el usuario puede realizar dos acciones: Consultar el estado de la tarea de beber en el día actual y consultar el histórico de tareas de beber realizadas en días anteriores.

### D.3.1. Consultar estado de la tarea

Para consultar el estado actual de la tarea de beber, el usuario debe dirigirse a la sección *líquido* de nuestra aplicación móvil y seguidamente, pulsar en la pestaña *consumo actual*.



Figura manuales 25.- Consultar estado de la tarea de beber

### D.3.2. Consultar histórico

El usuario puede consultar el histórico de tareas de beber realizadas días anteriores. Para ello, deberá pulsar en la pestaña *historial* dentro de la sección *líquido* de nuestra aplicación móvil y la aplicación le mostrara la información recogida en días anteriores.

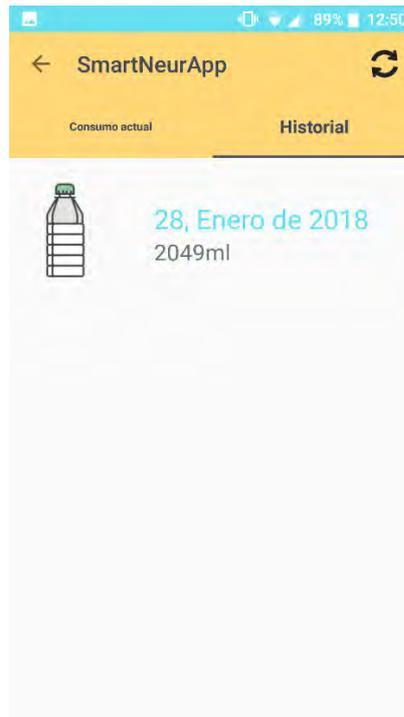


Figura manuales 26.- Histórico de las tareas de comer

## D.4. Cierre de sesión

El usuario puede cerrar la sesión en nuestra aplicación móvil para finalizar su uso. Para ello, debe dirigirse a la sección *perfil de usuario* de nuestra aplicación móvil y pulsar en la opción *cierre de sesión*.



Figura manuales 27. Perfil de usuario en la aplicación móvil

Seguidamente, el usuario deberá aceptar el cuadro de confirmación que nuestra aplicación le muestra y finalmente, nuestra aplicación de forma automática se cerrara.



Figura manuales 28.- Confirmación de cierre de sesión en aplicación móvil

## E. Manual de usuario de los sensores

### E.1. Uso del sensor de beber

Para usar nuestro sensor de comer, el usuario deberá colocarlo en posición horizontal sobre las paredes exteriores del vaso con el que se va a realizar las tareas de comer (véase en la siguiente figura).



Figura manuales 29.- Uso del sensor de beber

Finalmente, es importante destacar que se debe procurar que el sensor no entre en contacto con el líquido que se desea ingerir ya que puede suponer que los sensores puedan estropearse.

### E.2. Uso del sensor de comer

Para hacer uso de nuestros sensores en la realización de las tareas de comer por parte de los usuarios monitorizados, el usuario deberá colocar en sensor sobre una muñequera tal y como se muestra en la siguiente figura.

Además, no debe retirar el sensor de su muñeca hasta que no finalice su actividad de comer.



Figura manuales 30.- Uso del sensor de comer