

TRABAJO FIN DE GRADO



UCAM
UNIVERSIDAD CATÓLICA
DE MURCIA

ESCUELA UNIVERSITARIA POLITECNICA

Departamento de Ciencias Politécnicas
Grado en Ingeniería Informática

Aplicación web para la visualización de datos de sensor
“Smart Bin”

Autor: D. Alberto Pons Buttazzo

Directores: Dr. D. Fernando Terroso Sáenz
Dr. D. Andrés Muñoz Ortega

Murcia, Junio de 2021

TRABAJO FIN DE GRADO



UCAM

UNIVERSIDAD CATÓLICA
DE MURCIA

ESCUELA UNIVERSITARIA POLITECNICA

Departamento de Ciencias Politécnicas
Grado en Ingeniería Informática

Aplicación web para la visualización de datos de sensor
“Smart Bin”

Autor: D. Alberto Pons Buttazzo

Directores: Dr. D. Fernando Terroso Sáenz
Dr. D. Andrés Muñoz Ortega

Murcia, Junio de 2021

Dr. D. Fernando Terroso Sáenz y Dr. D. Andrés Muñoz Ortega profesores de la UCAM.

CERTIFICA: que el Trabajo Fin de Grado titulado *Aplicación web para la visualización de datos de sensor "Smart Bin"* que presenta D. Alberto Pons Buttazzo, para optar al título oficial de Grado en Ingeniería informática, ha sido realizado bajo su dirección.

A su juicio reúne las condiciones necesarias para ser presentado en la Universidad Católica San Antonio de Murcia y ser juzgado por el tribunal correspondiente.

Murcia, Junio del 2021

Agradecimientos

En este apartado me gustaría agradecer todo el apoyo que he recibido mientras realizaba el TFG. En primer lugar, quiero empezar por mi familia, que me ha apoyado en todo momento, y por ello le doy las gracias a mi padre, a mi madre y a mi hermano. En segundo lugar, mis agradecimientos van dirigidos a mis amigos, con los que me he podido divertir y dejar de vez en cuando el trabajo a un lado. Por último, quiero agradecer la tutela que he recibido durante el transcurso del TFG, especialmente a mis tutores D. Fernando Terroso Sáenz y D. Andrés Muñoz Ortega.

Muchísimas gracias a todos y cada uno de vosotros.

Contenido

| | |
|---|----|
| Resumen..... | 30 |
| Palabras clave..... | 30 |
| Abstract | 32 |
| Keywords..... | 32 |
| 1. Introducción | 34 |
| 1.1. Motivación | 34 |
| 1.2. Definición | 35 |
| 1.3. Objetivos propuestos | 35 |
| 2. Estado del arte y estudio de viabilidad | 37 |
| 2.1. Relación de proyectos con la misma funcionalidad | 37 |
| 2.2. Estudio de viabilidad | 39 |
| 2.2.1. Alcance del proyecto | 39 |
| 2.2.2. Estudio y valoración de las alternativas de solución..... | 40 |
| 2.2.3. Selección de la solución..... | 40 |
| 3. Metodologías usadas | 43 |
| 4. Tecnologías y herramientas utilizadas en el proyecto | 45 |
| 4.1. Herramientas software para la aplicación web..... | 45 |
| 4.2. Herramientas software para la aplicación inteligente | 46 |
| 4.3. Sensor de peso | 47 |
| 4.4. Herramientas de gestión..... | 48 |
| 5. Estimación de recursos y planificación | 49 |
| 5.1. Estimación de costes | 49 |
| 5.1.1. Definición del scrum team | 49 |
| 5.1.2. Definición del product backlog..... | 49 |
| 5.1.3. Identificación de los puntos de usuario | 52 |
| 5.1.4. Estimación de esfuerzo por historia de usuario | 52 |
| 5.1.5. Identificación de tareas, y estimación de esfuerzo y duración | 53 |
| 5.2. Planificación temporal del proyecto | 60 |
| 5.2.1. Definición de los sprints | 60 |
| 5.2.2. Diagrama temporal de sprints..... | 61 |
| 5.3. Valoración de la dedicación y el coste económico..... | 63 |
| 5.3.1. Valoración estimada de la dedicación y el coste económico..... | 63 |
| 5.3.2. Valoración real de la dedicación y el coste económico..... | 64 |
| 6. Desarrollo del contenido del proyecto..... | 65 |

| | | |
|--------|---|-----|
| 6.1. | Estructura básica de desarrollo..... | 65 |
| 6.2. | Sprint 1 - Base + Historia de usuario 1: Importación de CSV | 66 |
| 6.2.1. | Sprint 1 Planning Meeting..... | 66 |
| 6.2.2. | Sprint 1 Análisis mediante casos de uso | 67 |
| 6.2.3. | Sprint 1 Diseño mediante UML | 68 |
| 6.2.4. | Sprint 1 Desarrollo..... | 68 |
| 6.2.5. | Sprint 1 Review..... | 71 |
| 6.2.6. | Sprint 1 Retrospective | 72 |
| 6.3. | Sprint 2 - Historia de usuario 2: Visualización de datos | 74 |
| 6.3.1. | Sprint 2 Planning Meeting..... | 74 |
| 6.3.2. | Sprint 2 Análisis mediante casos de uso | 75 |
| 6.3.3. | Sprint 2 Diseño mediante UML | 75 |
| 6.3.4. | Sprint 2 Desarrollo..... | 76 |
| 6.3.5. | Sprint 2 Review..... | 78 |
| 6.3.6. | Sprint 2 Retrospective | 79 |
| 6.4. | Sprint 3 - 1/2 Historia de usuario 3: Generación de vistas..... | 79 |
| 6.4.1. | Sprint 3 Planning Meeting..... | 79 |
| 6.4.2. | Sprint 3 Análisis mediante casos de uso | 80 |
| 6.4.3. | Sprint 3 Diseño mediante UML | 81 |
| 6.4.4. | Sprint 3 Desarrollo..... | 82 |
| 6.4.5. | Sprint 3 Review..... | 85 |
| 6.4.6. | Sprint 3 Retrospective | 86 |
| 6.5. | Sprint 4 - Extras finales + 2/2 Historia de usuario 3: Generación de vistas | 86 |
| 6.5.1. | Sprint 4 Planning Meeting..... | 86 |
| 6.5.2. | Sprint 4 Análisis mediante casos de uso | 87 |
| 6.5.3. | Sprint 4 Diseño mediante UML | 88 |
| 6.5.4. | Sprint 4 Desarrollo..... | 89 |
| 6.5.5. | Sprint 4 Review..... | 90 |
| 6.5.6. | Sprint 4 Retrospective | 91 |
| 6.6. | Sprint 5 - Historia de usuario 4: Implementación de algoritmos | 92 |
| 6.6.1. | Sprint 5 Planning Meeting..... | 92 |
| 6.6.2. | Sprint 5 Desarrollo..... | 93 |
| 6.6.3. | Sprint 5 Review..... | 100 |
| 6.6.4. | Sprint 5 Retrospective | 100 |
| 6.7. | Sprint 6 - Historia de usuario 5: Generación de predicciones..... | 101 |
| 6.7.1. | Sprint 6 Planning Meeting..... | 101 |

| | | |
|---------|---|-----|
| 6.7.2. | Sprint 6 Desarrollo..... | 101 |
| 6.7.3. | Sprint 6 Review..... | 109 |
| 6.7.4. | Sprint 6 Retrospective | 109 |
| 6.8. | Sprint 7 - Historia de usuario 6: Integración de ambas aplicaciones | 110 |
| 6.8.1. | Sprint 7 Planning Meeting..... | 110 |
| 6.8.2. | Sprint 7 Análisis mediante casos de uso | 111 |
| 6.8.3. | Sprint 7 Diseño mediante UML | 112 |
| 6.8.4. | Sprint 7 Desarrollo..... | 113 |
| 6.8.5. | Sprint 7 Review..... | 115 |
| 6.8.6. | Sprint 7 Retrospective | 116 |
| 6.9. | Sprint 8 - Historia de usuario 7: Ajuste de curva en la aplicación inteligente..... | 116 |
| 6.9.1. | Sprint 8 Planning Meeting..... | 116 |
| 6.9.2. | Sprint 8 Análisis mediante casos de uso | 119 |
| 6.9.3. | Sprint 8 Diseño mediante UML | 119 |
| 6.9.4. | Sprint 8 Desarrollo | 120 |
| 6.9.5. | Sprint 8 Review..... | 122 |
| 6.9.6. | Sprint 8 Retrospective | 123 |
| 6.10. | Retrospectiva general de todo el desarrollo | 123 |
| 7. | Despliegue y prueba de la solución..... | 127 |
| 7.1. | Plan de Pruebas..... | 127 |
| 7.2. | Indicadores del desempeño y/o utilidad..... | 134 |
| 7.3. | Escalabilidad y mantenimiento | 135 |
| 8. | Conclusiones..... | 137 |
| 8.1. | Objetivos alcanzados..... | 137 |
| 8.2. | Conclusiones del trabajo | 139 |
| 8.3. | Vías futuras..... | 140 |
| 9. | Bibliografía | 141 |
| 10. | Anexos | 145 |
| 10.1. | Manual de instalación | 145 |
| 10.1.1. | Descarga e instalación de XAMPP | 145 |
| 10.1.2. | Descarga del código fuente y de los ficheros de instalación..... | 147 |
| 10.1.3. | Configuración de Apache | 148 |
| 10.1.4. | Configuración de phpMyAdmin | 150 |
| 10.1.5. | Despliegue de la web | 151 |
| 10.1.6. | Configuración del entorno Anaconda | 152 |
| 10.2. | Manual de usuario..... | 160 |

| | | |
|---------|--------------------------------|-----|
| 10.2.1. | Ventanas de la aplicación..... | 160 |
| 10.2.2. | Ventana de inicio..... | 161 |
| 10.2.3. | Ventana de vistas | 168 |
| 10.2.4. | Ventana de configuración | 169 |

Índice de ilustraciones

| | |
|---|-----|
| Ilustración 1. Relación entre objetivos generales y específicos..... | 36 |
| Ilustración 2. Comparativa entre Java y PHP (Guiadev, 2019)..... | 46 |
| Ilustración 3. Sensor de peso | 48 |
| Ilustración 4. Contenedor de plástico..... | 48 |
| Ilustración 5. Diagrama de Gantt Inicial..... | 62 |
| Ilustración 6. Sprint 1 - Sprint Backlog inicial..... | 67 |
| Ilustración 7. Diagrama casos de uso del Sprint 1..... | 67 |
| Ilustración 8. Diagrama UML del Sprint 1..... | 68 |
| Ilustración 9. Datos del CSV generado por el sensor de peso | 69 |
| Ilustración 10. Sprint 1 - Incremento..... | 72 |
| Ilustración 11. Burn Down Chart Sprint 1 | 72 |
| Ilustración 12. Sprint 2 - Sprint Backlog inicial..... | 74 |
| Ilustración 13. Diagrama casos de uso del Sprint 2..... | 75 |
| Ilustración 14. Diagrama UML del Sprint 2..... | 76 |
| Ilustración 15. Sprint 2 - Incremento..... | 78 |
| Ilustración 16. Burn Down Chart Sprint 2 | 79 |
| Ilustración 17. Sprint 3 - Sprint Backlog inicial..... | 80 |
| Ilustración 18. Diagrama casos de uso del Sprint 3..... | 81 |
| Ilustración 19. Diagrama UML del Sprint 3..... | 82 |
| Ilustración 20. Sprint 3 - Sprint Backlog intermedio | 84 |
| Ilustración 21. Sprint 3 - Incremento..... | 85 |
| Ilustración 22. Burn Down Chart Sprint 3 | 85 |
| Ilustración 23. Sprint 4 - Sprint Backlog inicial..... | 87 |
| Ilustración 24. Diagrama casos de uso del Sprint 3..... | 88 |
| Ilustración 25. Diagrama UML del Sprint 4..... | 89 |
| Ilustración 26. Sprint 4 - Incremento..... | 91 |
| Ilustración 27. Burn Down Chart Sprint 4 | 91 |
| Ilustración 28. Sprint 5 - Sprint Backlog inicial..... | 92 |
| Ilustración 29. Tabla con pronóstico de Prophet de los pesos medios..... | 94 |
| Ilustración 30. Gráfica con pronóstico de Prophet de los pesos medios | 95 |
| Ilustración 31. Pronóstico de Arima de los pesos medios | 96 |
| Ilustración 32. Dataframe con los pesos por cada instante..... | 97 |
| Ilustración 33. Dataframe con los pesos medios por cada segundo..... | 97 |
| Ilustración 34. Predicción con Prophet..... | 99 |
| Ilustración 35. Sprint 5 - Incremento..... | 100 |
| Ilustración 36. Burn Down Chart Sprint 5 | 100 |
| Ilustración 37. Sprint 6 - Sprint Backlog inicial..... | 101 |
| Ilustración 38. Predicción con Arima | 103 |
| Ilustración 39. Sprint 6 - Incremento..... | 109 |
| Ilustración 40. Burn Down Chart Sprint 6 | 109 |
| Ilustración 41. Sprint 7 - Sprint Backlog inicial..... | 111 |
| Ilustración 42. Diagrama casos de uso del Sprint 7..... | 112 |
| Ilustración 43. Diagrama UML del Sprint 7..... | 113 |
| Ilustración 44. Sprint 7 - Incremento..... | 115 |
| Ilustración 45. Burn Down Chart Sprint 7 | 116 |
| Ilustración 46. Sprint 8 - Sprint Backlog inicial..... | 118 |

| | |
|--|-----|
| Ilustración 47. Diagrama casos de uso del Sprint 8..... | 119 |
| Ilustración 48. Diagrama UML del Sprint 8..... | 120 |
| Ilustración 49. Tipos de curvas..... | 121 |
| Ilustración 50. Sprint 8 - Incremento..... | 122 |
| Ilustración 51. Burn Down Chart Sprint 8 | 123 |
| Ilustración 52. Gantt inicial | 124 |
| Ilustración 53. Gantt final..... | 125 |
| Ilustración 54. Burn Down Chart General..... | 125 |
| Ilustración 55. Diagrama casos de uso con pruebas..... | 128 |
| Ilustración 56. Instalación: descargar XAMPP..... | 145 |
| Ilustración 57. Instalación: instalar XAMPP 1 | 146 |
| Ilustración 58. Instalación: instalar XAMPP 2 | 147 |
| Ilustración 59. Instalación: descargar código fuente..... | 147 |
| Ilustración 60. Instalación: guardar ficheros..... | 148 |
| Ilustración 61. Instalación: modificar ficheros 1 | 148 |
| Ilustración 62. Instalación: modificar ficheros 2 | 149 |
| Ilustración 63. Instalación: iniciar Apache y MySQL..... | 149 |
| Ilustración 64. Instalación: crear tabla BD 1 | 150 |
| Ilustración 65. Instalación: crear tabla BD 2..... | 150 |
| Ilustración 66. Instalación: importar BD | 151 |
| Ilustración 67. Instalación: despliegue de la web | 151 |
| Ilustración 68. Instalación: descargar Anaconda Navigator..... | 152 |
| Ilustración 69. Instalación: instalar Anaconda Navigator 1 | 152 |
| Ilustración 70. Instalación: instalar Anaconda Navigator 2 | 153 |
| Ilustración 71. Instalación: ejecutar entorno root..... | 154 |
| Ilustración 72. Instalación: comando de modificación del entorno 1 | 155 |
| Ilustración 73. Instalación: comando de modificación del entorno 2.1..... | 155 |
| Ilustración 74. Instalación: comando de modificación del entorno 2.2..... | 156 |
| Ilustración 75. Instalación: comando de modificación del entorno 3.1..... | 156 |
| Ilustración 76. Instalación: comando de modificación del entorno 3.2..... | 157 |
| Ilustración 77. Instalación: comando de modificación del entorno 4.1..... | 157 |
| Ilustración 78. Instalación: comando de modificación del entorno 4.2..... | 158 |
| Ilustración 79. Instalación: comando de modificación del entorno 5.1..... | 158 |
| Ilustración 80. Instalación: comando de modificación del entorno 5.2..... | 159 |
| Ilustración 81. Instalación: modificar directorios 1..... | 159 |
| Ilustración 82. Instalación: modificar directorios 2..... | 160 |
| Ilustración 83. Uso: menú de navegación | 161 |
| Ilustración 84. Uso: cambiar de idioma..... | 161 |
| Ilustración 85. Uso: importar CSV..... | 161 |
| Ilustración 86. Uso: tabla con los registros..... | 162 |
| Ilustración 87. Uso: ordenar registros ascendentemente..... | 162 |
| Ilustración 88. Uso: buscar registros 1 | 163 |
| Ilustración 89. Uso: buscar registros 2..... | 163 |
| Ilustración 90. Uso: ver múltiples registros..... | 164 |
| Ilustración 91. Uso: gráfica con múltiples registros..... | 164 |
| Ilustración 92. Uso: gráfica de instante por peso | 165 |
| Ilustración 93. Uso: gráfica de instante por peso diferente..... | 165 |
| Ilustración 94. Uso: gráfica de instante por peso diferente (parte entera) | 165 |

| | |
|---|-----|
| Ilustración 95. Uso: gráfica de peso medio en franjas de 10 instantes..... | 166 |
| Ilustración 96. Uso: gráfica de peso medio en franjas de 20 instantes..... | 166 |
| Ilustración 97. Uso: gráfica de peso medio en franjas de 50 instantes..... | 166 |
| Ilustración 98. Uso: gráfica de peso medio en franjas de 100 instantes | 167 |
| Ilustración 99. Uso: gráfica de predicción con Prophet | 167 |
| Ilustración 100. Uso: gráfica de predicción con Prophet con ajuste de curva | 168 |
| Ilustración 101. . Uso: datos generales..... | 168 |
| Ilustración 102. Uso: gráficas con pesos medios | 169 |
| Ilustración 103. Uso: gráficas con pesos máximos | 169 |
| Ilustración 104: Uso: configuración de gráficas | 170 |
| Ilustración 105. Uso: zoom 1 | 170 |
| Ilustración 106. Uso: zoom 2 | 171 |
| Ilustración 107. Uso: exportación..... | 171 |

Índice de tablas

| | |
|---|----|
| Tabla 1. Comparación de proyectos similares..... | 39 |
| Tabla 2. Historia de usuario 1: Importación CSV | 50 |
| Tabla 3. Visualización de datos..... | 50 |
| Tabla 4. Generación de vistas..... | 51 |
| Tabla 5. Implementación de algoritmos | 51 |
| Tabla 6. Generación de predicciones..... | 51 |
| Tabla 7. Generación de predicciones..... | 51 |
| Tabla 8. Esfuerzo de los puntos de historia..... | 52 |
| Tabla 9. Estimación de esfuerzo por historia de usuario | 53 |
| Tabla 10. Tarea 1: Entorno de implementación para la web | 54 |
| Tabla 11. Tarea 2: BD para la web..... | 54 |
| Tabla 12. Tarea 3: MVC para la web | 54 |
| Tabla 13. Tarea 4: Diseño base para la web..... | 54 |
| Tabla 14. Tarea 5: Cargar CSV a BD para la web | 54 |
| Tabla 15. Tarea 6: Cargar registros a tabla para la web..... | 55 |
| Tabla 16. Tarea 7: Funcionalidades extras sobre la tabla para la web..... | 55 |
| Tabla 17. Tarea 8: Diseño sobre ventana de la tabla para la web | 55 |
| Tabla 18. Tarea 9: Funcionalidades de las vistas para la web | 55 |
| Tabla 19. Tarea 10: Diseño sobre ventana de las vistas para la web..... | 56 |
| Tabla 20. Tarea 11: Funcionalidades extras restantes para la web | 56 |
| Tabla 21. Tarea 12: Finalización de diseño para la web..... | 56 |
| Tabla 22. Esfuerzo y duración estimados en las historias de usuario 1, 2 y 3 | 56 |
| Tabla 23. Tarea 13: Entorno de implementación para la app inteligente | 57 |
| Tabla 24. Tarea 14: Investigación inicial de Machine Learning con Arima y Prophet para la app inteligente | 57 |
| Tabla 25. Tarea 15: Predicción de la variación de los pesos con Prophet para la app inteligente | 57 |
| Tabla 26. Tarea 16: Predicción de la variación de los pesos con Arima para la app inteligente | 58 |
| Tabla 27. Tarea 17: Predicción final de la variación de los pesos con Prophet para la app inteligente | 58 |
| Tabla 28. Esfuerzo y duración estimados en las historias de usuario 4 y 5 | 58 |
| Tabla 29. Tarea 18: Entorno de implementación para la integración..... | 58 |
| Tabla 30. Tarea 19: Conexión de ambas aplicaciones para la integración..... | 59 |
| Tabla 31. Tarea 20: Datos de entrada para la integración | 59 |
| Tabla 32. Tarea 21: Modificación de la app inteligente para la integración..... | 59 |
| Tabla 33. Tarea 22: Datos de salida para la integración | 59 |
| Tabla 34. Tarea 23: Generación de vistas para la integración | 60 |
| Tabla 35. Esfuerzo y duración estimados en las historia de usuario 6..... | 60 |
| Tabla 36. Asignación de los sprints | 61 |
| Tabla 37. Costes estimados del proyecto | 64 |
| Tabla 38. Costes reales del proyecto..... | 64 |
| Tabla 39. Sprint 1 Backlog | 67 |
| Tabla 40. Sprint 2 Backlog | 74 |
| Tabla 41. Tarea 9.1: Cambios en vistas y tabla | 80 |
| Tabla 42. Sprint 3 Backlog | 80 |
| Tabla 43. Internalización al inglés | 83 |

| | |
|--|-----|
| Tabla 44. Tarea 9.2: Internalización al inglés | 84 |
| Tabla 45. Sprint 3 Backlog intermedio | 84 |
| Tabla 46. Sprint 4 Backlog | 87 |
| Tabla 47. Sprint 5 Backlog | 92 |
| Tabla 48. Primeros resultados de Prophet..... | 99 |
| Tabla 49. Sprint 6 Backlog | 101 |
| Tabla 50. Primeros resultados de Arima | 103 |
| Tabla 51. Resultados finales de Prophet: registro eHMc1is7KT | 104 |
| Tabla 52. Resultados finales de Prophet: registro JmJ2por3Xk | 105 |
| Tabla 53. Resultados finales de Prophet: registro fAFmRC5wb7..... | 105 |
| Tabla 54. Resultados finales de Prophet: registro AHwGDvXm1Y..... | 106 |
| Tabla 55. Resultados finales de Prophet: registro S3cQCqyeJc | 106 |
| Tabla 56. Resultados finales de Prophet: registro GIhIPyhIFh | 107 |
| Tabla 57. Resultados finales de Prophet: registro kSqTilqCT4..... | 107 |
| Tabla 58. Resultados finales de Prophet: registro UDaUHwzMye..... | 108 |
| Tabla 59. Sprint 7 Backlog | 111 |
| Tabla 60. Generación de predicciones | 117 |
| Tabla 61. Tarea 24: Datos de entrada para la integración con ajuste de curva..... | 117 |
| Tabla 62. Tarea 25: Modificación de la app inteligente con ajuste de curva..... | 117 |
| Tabla 63. Tarea 26: Datos de salida para la integración con ajuste de curva | 117 |
| Tabla 64. Tarea 27: Generación de vistas para la integración con ajuste de curva | 118 |
| Tabla 65. Tarea 28: Carga con spinner para la aplicación inteligente..... | 118 |
| Tabla 66. Sprint 8 Backlog | 118 |
| Tabla 67. Pruebas unitarias..... | 134 |
| Tabla 68. Rendimiento mediante valores KPI | 135 |
| Tabla 69. Cumplimiento de objetivos | 139 |

RESUMEN

El problema que este proyecto trata de resolver responde a la optimización en la gestión de recogida de residuos. Se parte de la hipótesis de que la variación en el contenido de los desechos refleja un patrón de consumo por parte del ser humano.

El objetivo, para paliar dicho problema, consiste en estudiar los datos recogidos por un sensor de peso. Este se coloca sobre un contenedor de basura para seguir un registro de cómo varía su contenido con el tiempo. En base a los estudios realizados, se busca reducir tiempos y costes durante la recolección de residuos. Además, esto puede evitar acumulaciones de basura, repercutiendo de forma positiva en el medio ambiente.

Para ello se ha creado una aplicación web que permite la visualización de los datos que el sensor de peso registra. A fin de analizar dichos datos, se ha creado también una aplicación inteligente que hace uso de técnicas de Machine Learning y de la librería Prophet.

Las respuestas que se han obtenido son patrones de comportamiento en la variación de peso de desechos en los contenedores. Con estos hallazgos, se concluye que se puede generar una repuesta más eficiente en la estrategia de planificación de recogida, usando dichos patrones.

PALABRAS CLAVE

Residuos, Machine Learning, Patrón de comportamiento, Aplicación web, Visualización de datos, Prophet.

Aplicación web para la visualización de datos de sensor "Smart Bin"

ABSTRACT

The problem that this project tries to solve responds to the optimization of waste collection management. Starting from the hypothesis that the variation in the content of the rubbish reflects a pattern of consumption by the human being.

The objective, to alleviate this problem, is to study the data collected by a weight sensor. It is placed on top of a garbage container to keep track of how its contents change over time. Based on the studies carried out, it seeks to reduce time and costs during waste collection. In addition, this can prevent accumulations of garbage, having a positive impact on the environment.

In order to achieve that, a web application that allows the visualization of the data that the weight sensor records has been created. In order to analyze this data, an intelligent application that makes use of Machine Learning techniques and Prophet library has also been created.

The results that have been obtained are behavior patterns in the variation of the weight of waste in the containers. Based on these findings, it is concluded that a more efficient response can be generated in the collection planning strategy, using these patterns.

KEYWORDS

Waste, Machine Learning, Behavior pattern, Web application, Visualization of data, Prophet.

1. INTRODUCCIÓN

1.1. MOTIVACIÓN

El presente estudio se enmarca en un proyecto desarrollado por el grupo de investigación UKEIM del grado en ingeniería informática de la UCAM, con la motivación de crear un sistema inteligente de recogida de basuras. En el proyecto se usan varios sensores para recoger datos de los contenedores, como por ejemplo, sensores de peso (para calcular la variación en el peso de los contenedores) o sensores infrarrojos (para calcular el nivel de llenado de los contenedores). De esta manera se obtiene información de gran valor para predecir patrones de comportamiento de los usuarios durante el depósito de sus residuos. En este contexto, el desarrollo de este TFG trata sobre la elaboración de una aplicación web que permita visualizar los datos de un sensor de peso, para ser posteriormente analizados mediante técnicas de Machine Learning.

El proyecto trata de buscar una mejora en la recogida de residuos, es decir, optimizar la estrategia de recolección cumpliendo dos requisitos:

- Conocer los pesos de los contenedores que se manejan.
- Buscar patrones de comportamiento de los usuarios en la variación de los pesos, como por ejemplo, sería el caso de una persona dependiente que no pueda depositar la basura fuera de casa por sí misma.

Esta optimización se consigue estudiando los cambios de contenido que sufren los contenedores, pero realmente el punto más relevante del estudio es analizar los patrones de comportamiento de los seres humanos al depositar sus desechos.

Con todo ello, el proyecto aborda el problema de la poca eficiencia a la hora de recolectar los residuos de los contenedores. Este suele ser un proceso que se realiza una vez al día, a la misma hora y siguiendo siempre el mismo recorrido. Esto puede generar mayores costes y un inadecuado uso de los tiempos utilizados, ya que no se sigue una estrategia coordinada al patrón de

consumo. Además puede generar acumulaciones de basura, lo que afecta negativamente al medio ambiente.

1.2. DEFINICIÓN

Una de las acciones para abordar el problema comentado anteriormente consiste en recoger datos del peso actual del contenedor mediante un sensor, para posteriormente poder ser analizados mediante técnicas de Machine Learning. Por tanto, el objetivo sería encontrar los patrones de comportamiento necesarios para optimizar la planificación de recogida.

Dentro de las variables recogidas por el sensor que se van a estudiar, están las siguientes:

- El tiempo de registro, que servirá para optimizar cuándo se va a realizar la recogida.
- Los pesos, que servirán para marcar una prioridad a los contenedores con mayor porcentaje de uso.

Analizando dichos datos se busca encontrar un patrón de comportamiento del que partirá la estrategia de recolección de residuos. Entre las tecnologías más destacables que se van a usar para buscar un patrón, están Prophet y Arima, dos librerías para generar predicciones, y que se explicarán con mayor detenimiento en el punto 4.2 (*Herramientas software para la aplicación inteligente*).

1.3. OBJETIVOS PROPUESTOS

Se desea conseguir los siguientes objetivos generales:

- OG1: Desarrollar una aplicación web que permita la visualización de los datos de un sensor de peso aplicado a un contenedor de plástico.
- OG2: Desarrollar una aplicación inteligente que hace uso de técnicas de Machine Learning para analizar los datos registrados por el sensor de peso.

El primer objetivo general se desglosa en dos objetivos específicos:

- OE1.1: Procesar ficheros en formato CSV (comma-separated values) y contener un repositorio con los valores que se obtienen de los distintos experimentos que se realicen.
- OE1.2: Generar diferentes vistas (gráficos, tablas, etc.) de los datos recogidos para ofrecer información relevante de los mismos.

El segundo objetivo general se desglosa también en dos objetivos específicos:

- OE2.1: Evaluar un conjunto de algoritmos que permitan encontrar patrones de variación de peso con fiabilidad.
- OE2.2: Hacer uso de los algoritmos evaluados en OE2.1 para asegurar un gran porcentaje de precisión en las futuras variaciones de los desechos.

La relación entre los objetivos generales y específicos se resume en la ilustración 1.

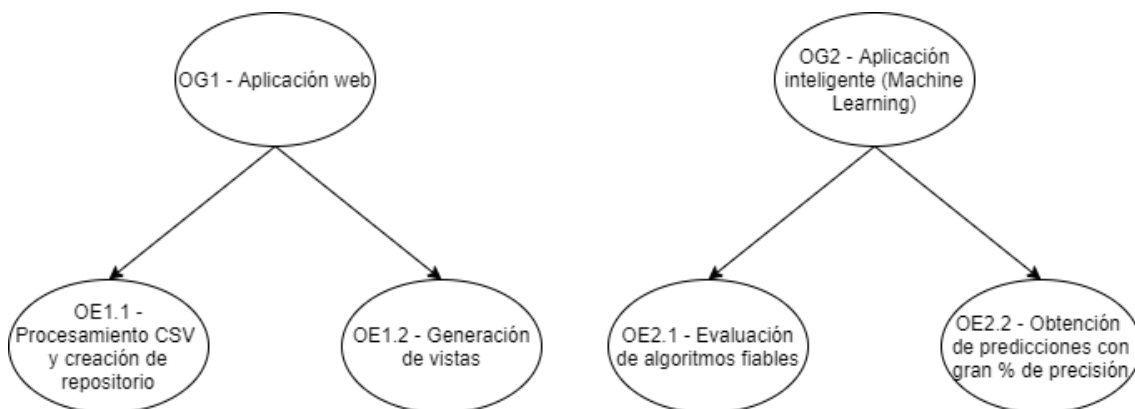


Ilustración 1. Relación entre objetivos generales y específicos

2. ESTADO DEL ARTE Y ESTUDIO DE VIABILIDAD

2.1. RELACIÓN DE PROYECTOS CON LA MISMA FUNCIONALIDAD

Existen proyectos parecidos a lo que se pretende realizar en este estudio. Se van a describir cuatro proyectos similares, que diseñan un sistema de monitorización de basura usando un sensor de peso.

En primer lugar, se va a exponer el proyecto *Design of IoT Garbage Monitoring with Weight Sensing* realizado por binti Md. Ishak, *et al* (binti Md. Ishak & bin Abdullah, 2020). Este trabajo consigue monitorizar el vaciado de los contenedores a través de una web, para optimizar el proceso de recogida. Hace uso de sensores ultrasónicos (para medir el nivel de llenado), de sensores de peso, de una pantalla LCD (para mostrar el nivel de contenido) y de un módem Wi-fi para notificar los datos recogidos. Con esto se consigue reducir el número de viajes de recolección, así como los costes globales para dichas tareas.

En segundo lugar, el proyecto *Smart Garbage Monitoring System Using Sensors With RFID Over Internet Of Things* realizado por Dhana Satyamanikanta, *et al* (Dhana Satyamanikanta & Madeshan, 2017). Los autores utilizan sensores infrarrojos, sensores de peso, sensores fotoeléctricos y un lector de tarjeta de identificación por radiofrecuencia (RFID) (permite la identificación de personas que lleven una tarjeta RFID al acercarse al contenedor, para posteriormente recibir información del contenido). Este estudio concluye que se obtienen unos mejores resultados al usar RFID sobre IoT, consiguiendo eliminar los desechos lo antes posible sin afectar al entorno.

En tercer lugar, el trabajo *IOT based Garbage Monitoring Using Arduino* realizado por Microtronics Technologies (Microtronics Technologies, 2020). Sigue un objetivo similar a los anteriores proyectos, intentando evitar acumulaciones de residuos. Hace uso de sensores de peso, sensores ultrasónicos, un módulo IoT, una pantalla LCD, y un mini controlador Arduino. Este último, cuando detecta que se ha alcanzado cierto umbral en el contenedor, envía notificaciones a la página web mediante el módulo IoT.

Por último, el proyecto *Waste Management System Using IoT-Based Machine Learning in University* realizado por Ahn Khoa, *et al* (Ahn Khoa, y otros,

2020). Este trabajo optimiza la recolección de residuos en una universidad. Además de usar sensores de peso y ultrasónicos, usa un módulo LoRa para transferir los datos recogidos a un servidor. Pero lo realmente novedoso, en comparación con los anteriores proyectos, es el uso de Machine Learning y teoría de grafos. Con esto último, se consigue ahorrar tiempo al identificar la ruta más óptima a la hora de gestionar la recogida de desechos.

Como se ha podido observar durante las exposiciones que se resumen en la tabla 1, la mayoría de estudios implementan la visualización de los datos recogidos a través de distintos sensores. Y el último proyecto además implementa el uso de técnicas de Machine Learning para predecir las mejores rutas de planificación de recogida. Este estudio pretende realizar algo parecido, pudiendo diferenciarse en el nivel de rigidez al presentar los datos y el nivel de predicción que se quiere obtener.

| Proyecto | Recolección de datos | Visualización de datos | Análisis de los datos |
|--|--|---|------------------------------|
| <i>Design of IoT Garbage Monitoring with Weight Sensing</i> | Sensores ultrasónicos, sensores de peso y un módem Wi-fi | Página web y pantalla LCD | - |
| <i>Smart Garbage Monitoring System Using Sensors With RFID Over Internet Of Things</i> | Sensores infrarrojos, sensores de peso, sensores fotoeléctricos y RFID | Notificación por las pantallas de los oficiales | - |
| <i>IOT based Garbage Monitoring Using Arduino</i> | Sensores de peso, sensores ultrasónicos, un módulo IoT y un mini controlador Arduino | Página web y pantalla LCD | - |

| | | | |
|--|--|------------------|--|
| <i>Waste Management System Using IoT-Based Machine Learning in University</i> | Sensores de peso, sensores ultrasónicos y un módulo LoRa | Aplicación móvil | Machine Learning (Regresión logística) y teoría de grafos (Dijkstra) |
| <i>Aplicación web para la visualización de datos de sensor “Smart Bin” (TFG)</i> | Sensor de peso | Página web | Machine Learning (usando las librerías Prophet y Arima) |

Tabla 1. Comparación de proyectos similares

2.2. ESTUDIO DE VIABILIDAD

2.2.1. ALCANCE DEL PROYECTO

El proyecto forma parte de un sistema inteligente de recogida de residuos que usa diferentes sensores para recoger datos de los contenedores (pesos, niveles de llenado, etc.). Con el objetivo de obtener información valiosa que ayude a predecir patrones de comportamiento de los usuarios durante el depósito de sus residuos. El alcance del TFG está relacionado con la gestión de un sensor de peso de ese sistema. Se pretende desarrollar una aplicación web que permita la visualización de los datos de dicho sensor y desarrollar una aplicación inteligente, que hace uso de técnicas de Machine Learning para analizar los datos recogidos.

El proyecto tiene como potenciales destinatarios:

- Personas dependientes que no puedan depositar la basura fuera de casa por sí mismas. Se trata de solventar sus necesidades de depósito de desechos, sin que se vean obligadas a acumular su basura dentro del hogar.

- Sistema administrativo de recogida de residuos. Se busca optimizar la planificación de recogida, reducir costes, tiempos e acumulaciones de residuos.
- Sistema administrativo de recolección de residuos sanitarios. Con el propósito de mantener la higiene al evitar acumulaciones de material sanitario.

2.2.2. ESTUDIO Y VALORACIÓN DE LAS ALTERNATIVAS DE SOLUCIÓN

Los distintos caminos que se pueden seguir para realizar el proyecto vienen definidos por:

- Nivel de profundidad en la recolección de datos: si se comparan los proyectos expuestos anteriormente en la sección 2.1 (*Relación con proyectos con la misma funcionalidad*), se puede ver que hay una gran variedad de sensores a la hora de recolectar datos de los contenedores. Como por ejemplo, sensores fotoeléctricos, ultrasónicos, de peso o infrarrojos. Además estos sensores transfieren los datos gracias a tecnologías tales como, módulos LoRa, Wi-fi, IoT, o controladores Arduino.
- Nivel de profundidad en la visualización de los datos: cada proyecto presenta los datos de distinta forma, optando por una aplicación web u otros soportes cómo una pantalla LCD. Estos se muestran en forma de tabla, gráfica o mediante representaciones visuales de llenado.
- Nivel de análisis de los datos: aunque todos tratan de buscar patrones en la recogida de residuos, el último proyecto presentado es el único que plantea un análisis profundo de predicción mediante técnicas de Machine Learning y teoría de grafos.

2.2.3. SELECCIÓN DE LA SOLUCIÓN

En este proyecto se quiere seguir una trayectoria que sea viable al tiempo y esfuerzo que se debe dedicar, siguiendo las especificaciones establecidas por los objetivos que se pretenden conseguir:

Aplicación web para la visualización de datos de sensor “Smart Bin”

- Nivel de profundidad en la recolección de datos: se pretende recoger datos de un solo tipo de sensor, un sensor de peso.
- Nivel de profundidad en la visualización de los datos: se pretenden generar distintas vistas para visualizar los datos (tablas, gráficos, etc.) a través de una aplicación web.
- Nivel de análisis de los datos: se pretende crear una aplicación inteligente que hace uso de técnicas de Machine Learning para analizar los datos recogidos, tratando de obtener un porcentaje que prediga con fiabilidad la variación de los pesos en el tiempo.

Se considera que los niveles de profundidad elegidos son acordes al esfuerzo y tiempo que se le debe dedicar al TFG. A primera vista, se considera posible y viable alcanzar dichos objetivos. Aunque no se puede garantizar el cumplimiento de los mismos, al igual que no se puede asegurar que estos puedan sufrir extensiones para valorizar aún más el estudio.

3. METODOLOGÍAS USADAS

Antes de comparar las distintas metodologías a estudiar para su implementación, es preciso destacar que se va a partir de un enfoque ágil. Esto se ha decidido en base a las características del proyecto. Se entiende que al usar una metodología ágil, se consigue una mayor rapidez, dinamismo y capacidad de adaptación a cambios durante la elaboración del proyecto (Escuela de negocios EBF, 2019). Al partir de un trabajo que no es propuesto por el alumno, existe cierta inestabilidad en el entendimiento de la arquitectura inicial del proyecto. Esto se debe a que la idea no parte del alumno, y por tanto no existe un estudio profundo y previo a la realización del trabajo. Más bien, este estudio converge de forma paralela al desarrollo del proyecto.

Las metodologías tradicionales marcan muy detalladamente los requerimientos desde el principio, los cambios son más difíciles de implementar y no suelen tener un aspecto retroalimentativo con el cliente, y además se magnifica la documentación requerida.

Por todos los motivos anteriormente expuestos, se entiende que en el proyecto es necesario cierto margen de fallo y flexibilidad, lo cual se correlaciona muy bien con el uso de metodologías ágiles. Entre las metodologías a analizar están Scrum, programación extrema (XP), Kanban y Scrumban.

Empezando con Scrum (Izquierdo, Gómez, & Nava, 2019), esta metodología busca fraccionar el proyecto en etapas de reducido tamaño, lo que se conoce como Sprints. En estos se realizan reuniones diariamente (Daily Scrum) y revisiones semanalmente o mensualmente (Sprint Review Meeting), con lo que se ofrece un elevado grado de adaptación. Además existe una reunión inicial para planificar el proyecto (Sprint Planning Meeting) y reuniones para analizar la retrospectiva (Sprint Retrospective), es decir que se estudia el enfoque inicial y final del sprint. Se ofrecen ventajas tales como, flexibilidad, organización, productividad, reducción de riesgos o testeo del cliente antes que finalizar el proyecto.

Siguiendo con Kanban (Amor García, Guardiola Elías, & Marín García, 2019), es una metodología que busca organizar un proyecto mediante un panel

con distintas columnas (TO DO, DOING y DONE). Las tareas que se van elaborando para desarrollar el trabajo se les conoce como WIP (Work in progress). Estas se reparten por las columnas según su estado, y de arriba abajo según su prioridad. Se ofrecen ventajas tales como, adaptación al cambio, control y comprensión de la información, y capacidad de integración.

Siguiendo con XP (Parrales Cáceres, Sepúlveda Madrid, Fernández Segura, & Parra Navarro, 2019), es una metodología que se constituye a partir de cinco valores, simplicidad, comunicación, coraje, respeto y retroalimentación. Se establece una fase de planificación (historias de usuario y plan de interacción), diseño (simple con el uso de tarjetas CRC en el brainstorming), desarrollo (programación en parejas, ultimando las optimizaciones) y pruebas (corrección de errores con participación del cliente). Con ello se consigue una gran adaptabilidad, acortamiento de tiempos, sencillez en el código, reducción de errores y un fomento de comunicación entre desarrolladores y cliente.

Finalmente, Scrumban es una mezcla de Scrum y Kanban (Gálvez Sánchez, Torres Segura, Saura Cuadrado, & Pascoe-Kelly, 2019). Por un lado, estructura el trabajo también mediante columnas y mediante un WIP (Kanban). Y por otro lado, establece un nivel de prioridades, una serie de reuniones a demanda y revisiones (Scrum). Con lo anterior, se intenta flexibilizar el proceso de desarrollo, evitando la pérdida de tiempo.

Entre las metodologías descritas, se decide optar por Scrum en base a los siguientes motivos:

- Simplificar la elaboración del proyecto mediante el fraccionamiento del mismo en etapas (Sprints).
- Mantener un seguimiento continuo para evitar que los errores se vayan arrastrando durante el proyecto (reuniones y revisiones).
- Ofrecer cierto grado de flexibilidad y adaptación, para poder reaccionar ante cambios, alterando lo menos posible los costes y tiempos del desarrollo.

4. TECNOLOGÍAS Y HERRAMIENTAS UTILIZADAS EN EL PROYECTO

En este apartado se van a exponer los motivos por los cuales se van a utilizar unas tecnologías y herramientas, y por qué no otras.

4.1. HERRAMIENTAS SOFTWARE PARA LA APLICACIÓN WEB

En primer lugar, se van a justificar las tecnologías y herramientas seleccionadas para alcanzar el primer objetivo general (desarrollar una aplicación web para mostrar los datos recogidos por el sensor).

A nivel de software, se han estudiado las opciones de usar Java o PHP (Guiadev, 2019). Java es un lenguaje de programación orientado a objetos, que puede ejecutarse en múltiples plataformas y se puede usar gratuitamente. El lenguaje ofrece un buen rendimiento, gran portabilidad, seguridad y soporte. Aunque puede conllevar mayores costes y tiempo en el desarrollo que PHP. PHP es un lenguaje de programación libre y especializado en el campo del desarrollo de aplicaciones web. Este ofrece las mismas ventajas que Java, mitiga con efectividad las desventajas mencionadas, y ofrece una mayor simplicidad sintáctica y de aprendizaje. Aunque a nivel de seguridad Java es más competente que PHP, al integrar opciones de seguridad más avanzadas. Ambos lenguajes ofrecen una extensa documentación, lo que concede una gran ayuda a los desarrolladores.

Finalmente se ha decidido optar por PHP, debido a que ofrece mayores ventajas que Java, está especializado en el desarrollo web, y ya que no se considera que se esté trabajando con datos de gran sensibilidad que necesiten de una gran seguridad. Además, se va a acompañar este lenguaje junto a las herramientas phpMyAdmin (administración de MySQL) (phpMyAdmin, 2020), Visual Studio Code (editor de código fuente) (Microsoft, 2020), y XAMPP para gestionar el servidor web (Apache) (Apache Friends, 2020) y la base de datos. Se ha decidido seleccionar dichas tecnologías, ya que ofrecen una gran compatibilidad con PHP y son totalmente gratis.

Las ventajas y desventajas de ambos lenguajes finalmente se resumen en la ilustración 2.

| | PHP | Java |
|----------------------|------------------------|------------------------|
| Costos | ✔ Más barato | ✘ Más Costoso |
| Rendimiento | ✔ Buen Rendimiento | ✔ Buen Rendimiento |
| Seguridad | ✘ Menos seguro | ✔ Más seguro |
| Sintaxis | ✔ Amigable | ✘ Compleja |
| Portabilidad | ✔ Portable | ✔ Portable |
| Tiempo de Desarrollo | ✔ Poco tiempo | ✘ Más tiempo |
| Soporte y Ayuda | ✔ Buen Soporte y Ayuda | ✔ Buen Soporte y Ayuda |

Ilustración 2. Comparativa entre Java y PHP (Guiadev, 2019)

4.2. HERRAMIENTAS SOFTWARE PARA LA APLICACIÓN INTELIGENTE

En segundo lugar, se van a justificar las tecnologías y herramientas seleccionadas para alcanzar el segundo objetivo general (desarrollar una aplicación inteligente para analizar los datos recogidos por el sensor).

A nivel de herramientas de análisis de datos, se han examinado dos alternativas, R y Python, dos lenguajes que se compaginan muy bien con el análisis estadístico y científico (Guru99, 2020). Ambos ofrecen gran variedad de librerías que permiten gestionar grandes cantidades de datos y que se pueden usar gratuitamente. Aunque distan en el usuario objetivo, en la curva de dificultad y en el nivel de integración. Por estas diferencias, se ha elegido Python, ya que está más orientado a los programadores, ofrece una curva de aprendizaje más suave y permite una buena integración con otras aplicaciones.

Por otro lado, se ha decidido investigar el uso de las librerías Prophet (Taylor & Letham, 2017) y Arima (Brownlee, 2017) para predecir la variación de los pesos en el tiempo. La primera opción es un procedimiento que permite el pronóstico de datos que se organizan en una serie de tiempo (diaria, semanal y/o anual). La segunda opción, al igual que la primera, pronostica datos organizados en una serie temporal, aunque usa métodos estadísticos de variaciones y regresiones.

Respecto a los entornos de desarrollo que se van a usar, se ha decidido seleccionar Jupyter Notebook (Project Jupyter, 2021) y Google Colab (Google, 2021). Ambos entornos ofrecen una vía para analizar datos de manera científica y mediante Machine Learning. Cabe destacar, que se va a trabajar de manera online en ambos, para evitar sobrecargar el ordenador de un mayor número de aplicaciones.

Finalmente, se va a usar la aplicación Anaconda Navigator, que permite de forma gráfica y sin usar la línea de comandos, manejar paquetes, entornos y canales (Anaconda, 2021). Esta última herramienta permite que se pueda realizar la integración de la aplicación inteligente con la web, ya que facilita la descarga del entorno con las librerías que se van a necesitar de forma local.

4.3. SENSOR DE PESO

Finalmente, para el cumplimiento de ambos objetivos, es necesario capturar los datos que se mostrarán y analizarán mediante un sensor.

A nivel de hardware, este proyecto abarca la gestión de un sensor de peso. El sensor que se va a usar en este TFG se llama *Acaia Brew Scale* y posee las siguientes características (Acaia, 2020):

- Rapidez de respuesta al leer (20 ms).
- Gran precisión y sensibilidad (0.1 g).
- Capacidad de registrar el tiempo de las tomas.
- Gran estabilidad mediante un compensador de deslizamiento.
- Automatización del tiempo de apagado.
- Fuente de alimentación por USB y con capacidad de batería de 20 a 30 horas.

Por las características expuestas, es una alternativa hardware ideal para elaborar el proyecto. Permite registrar con precisión los datos, para luego poder ser analizados con la menor tasa de error posible. Se puede observar el sensor y el contenedor de plástico donde se utiliza en la ilustración 3 e ilustración 4 respectivamente.



Ilustración 3. Sensor de peso



Ilustración 4. Contenedor de plástico

4.4. HERRAMIENTAS DE GESTIÓN

También van a ser necesarias herramientas de apoyo, como Excel para la realización de diferentes gráficas (Microsoft, 2021), TeamGantt para realizar los diagramas de Gantt (TeamGantt, 2021) o Monday.com para definir los Product Backlog y los incrementos (Monday.com, 2021).

5. ESTIMACIÓN DE RECURSOS Y PLANIFICACIÓN

5.1. ESTIMACIÓN DE COSTES

5.1.1. DEFINICIÓN DEL SCRUM TEAM

El Scrum Team es el equipo que va a formar parte del desarrollo de ambas aplicaciones. Para cumplir una planificación acorde a Scrum, es necesario dividir una serie de tareas entre los tres integrantes del equipo:

- Product Owner, rol que corresponde a Dr. D. Andrés Muñoz Ortega, siendo el dueño de la aplicación que expresa los requerimientos que deben tener las aplicaciones, y asegura que el equipo de desarrollo los entiendan.
- Scrum Master, rol que corresponde a Dr. D. Fernando Terroso Sáenz, siendo el líder del proyecto que asegura que se siguen las prácticas y principios de Scrum en el proyecto.
- Development Team, integrado por Alberto Pons Buttazzo, siendo el único miembro del equipo de desarrollo que tiene el deber de crear y mantener el software.

5.1.2. DEFINICIÓN DEL PRODUCT BACKLOG

Antes de poder realizar las estimaciones de esfuerzo y duración de cada tarea, es necesario definir el Product Backlog, es decir, identificar las historias de usuario (requisitos del sistema desde el punto de vista del usuario). Durante la recogida de requisitos, se han identificado las siguientes historias de usuario que son reflejadas de la tabla 2 a la tabla 7:

- Como un cliente, quiero importar ficheros CSV de un sensor de peso a una aplicación web para que los datos se carguen en la página.
- Como un cliente, quiero visualizar los datos recogidos por el sensor de peso a través de una tabla para tener un control sobre lo que se registra.

- Como un cliente, quiero ver distintas vistas (gráficos principalmente) para enriquecer la información recogida por el sensor.
- Como un cliente, quiero que se implementen algoritmos de Machine Learning para analizar los pesos registrados.
- Como un cliente, quiero que se generen predicciones de gran fiabilidad en la variación de los pesos.
- Como un cliente, quiero que se integren la aplicación inteligente y la aplicación web.

| Historia de usuario 1: Importación CSV | |
|---|--|
| Rol | Cliente |
| Necesidad y beneficio | Importar ficheros CSV de un sensor de peso a una aplicación web para que los datos se carguen en la página |
| Criterio de aceptación (SMART) | Se cargan la totalidad de los datos y se mantiene la integridad de los mismos |

Tabla 2. Historia de usuario 1: Importación CSV

| Historia de usuario 2: Visualización de datos | |
|--|---|
| Rol | Cliente |
| Necesidad y beneficio | Visualizar los datos recogidos por el sensor de peso a través de una tabla para tener un control sobre lo que se registra |
| Criterio de aceptación (SMART) | Se cargan a la tabla la totalidad de los datos y se mantiene la integridad de los mismos |

Tabla 3. Visualización de datos

| Historia de usuario 3: Generación de vistas | |
|--|---------|
| Rol | Cliente |

| | |
|--------------------------------|--|
| Necesidad y beneficio | Ver distintas vistas (gráficos principalmente) para enriquecer la información recogida por el sensor |
| Criterio de aceptación (SMART) | Se cargan a las vistas los datos de forma íntegra |

Tabla 4. Generación de vistas

| Historia de usuario 4: Implementación de algoritmos | |
|--|--|
| Rol | Cliente |
| Necesidad y beneficio | Implementar algoritmos de Machine Learning para analizar los pesos registrados |
| Criterio de aceptación (SMART) | A criterio del desarrollador |

Tabla 5. Implementación de algoritmos

| Historia de usuario 5: Generación de predicciones | |
|--|---|
| Rol | Cliente |
| Necesidad y beneficio | Generar predicciones de gran fiabilidad en la variación de los pesos |
| Criterio de aceptación (SMART) | Las predicciones deben ser fiables, es decir, que la precisión de las mismas deberá ser justificada mediante métricas |

Tabla 6. Generación de predicciones

| Historia de usuario 6: Integración de ambas aplicaciones | |
|---|--|
| Rol | Cliente |
| Necesidad y beneficio | Integrar la aplicación inteligente y la aplicación web |
| Criterio de aceptación (SMART) | La aplicación web permite usar la aplicación inteligente |

Tabla 7. Generación de predicciones

Una vez identificados los puntos de historia, se debe estimar la complejidad o esfuerzo de cada uno de ellos.

5.1.3. IDENTIFICACIÓN DE LOS PUNTOS DE USUARIO

Para elaborar una estimación de esfuerzo se ha decidido usar el método Planning Poker (Kzi-Kaizenia, 2020). Un requisito previo para usar esta técnica consiste en identificar los puntos de historia (tamaño de una historia de usuario). Luego, el método se ejecuta asignando a cada historia de usuario un punto de historia. En este caso, el puntaje de las historias va a seguir la serie de Fibonacci. Se ha decidido usar este método, ya que, simplifica la estimación de cada entrada del Product Backlog, permite participar a los clientes con mayor facilidad e interactividad, y porque es una técnica muy extendida en las planificaciones ágiles, y particularmente en Scrum. Se han identificado los siguientes puntos de historia en la tabla 8:

| Puntos de historia | Complejidad/Esfuerzo |
|---------------------------|-----------------------------|
| 0 | 1 día |
| 1 | 3 días |
| 2 | 5 días |
| 3 | 1 semana |
| 5 | 2 semanas |
| 8 | 3 semanas |
| 13 | 1 mes |

Tabla 8. Esfuerzo de los puntos de historia

5.1.4. ESTIMACIÓN DE ESFUERZO POR HISTORIA DE USUARIO

A continuación, se asigna a cada historia de usuario un punto de historia en la tabla 9:

| ID | Valor cliente | Historia de usuario | Puntos |
|-----------|----------------------|---|---------------|
| 01 | IMP 3 | Historia de usuario 1: Importación CSV | 1 |
| 02 | TOP 2 | Historia de usuario 2: | 2 |

| | | | |
|----|-------|---|----|
| | | Visualización de datos | |
| 03 | IMP 5 | Historia de usuario 3: Generación de vistas | 13 |
| 04 | IMP 4 | Historia de usuario 4: Implementación de algoritmos | 5 |
| 05 | TOP 1 | Historia de usuario 5: Generación de predicciones | 3 |
| 06 | IMP 6 | Historia de usuario 6: Integración de ambas aplicaciones | 8 |

Tabla 9. Estimación de esfuerzo por historia de usuario

Se puede observar que también se tiene en cuenta el orden de prioridad del cliente (*Valor cliente*).

5.1.5. IDENTIFICACIÓN DE TAREAS, Y ESTIMACIÓN DE ESFUERZO Y DURACIÓN

A partir de las historias de usuario se identifican las siguientes tareas de la tabla 10 a la tabla 35:

| | | |
|--|------------------------|---|
| Nombre del integrante: Alberto Pons Buttazzo | | |
| Descripción de la actividad: preparar el entorno de implementación para empezar a desarrollar la aplicación web. Adquirir y configurar herramientas que se van a usar (XAMPP y Visual Studio Code). | | |
| Duración: 1 día | Esfuerzo: 1 día | Necesario para historias de usuario: 1, 2 y 3 |

Tabla 10. Tarea 1: Entorno de implementación para la web

| | | |
|---|------------------------|---|
| Nombre del integrante: Alberto Pons Buttazzo | | |
| Descripción de la actividad: preparar base de datos con phpMyAdmin. Generar tablas y transacciones necesarias. | | |
| Duración: 1 día | Esfuerzo: 1 día | Necesario para historias de usuario: 1, 2 y 3 |

Tabla 11. Tarea 2: BD para la web

| | | |
|---|-------------------------|---|
| Nombre del integrante: Alberto Pons Buttazzo | | |
| Descripción de la actividad: generar estructura base Modelo vista controlador (MVC). | | |
| Duración: 4 días | Esfuerzo: 3 días | Necesario para historias de usuario: 1, 2 y 3 |

Tabla 12. Tarea 3: MVC para la web

| | | |
|--|-------------------------|---|
| Nombre del integrante: Alberto Pons Buttazzo | | |
| Descripción de la actividad: implementar diseño base de la aplicación web (cabecera, cuerpo y pie de página). | | |
| Duración: 4 días | Esfuerzo: 3 días | Necesario para historias de usuario: 1, 2 y 3 |

Tabla 13. Tarea 4: Diseño base para la web

| | | |
|---|-------------------------|--|
| Nombre del integrante: Alberto Pons Buttazzo | | |
| Descripción de la actividad: implementar funcionalidad para cargar los CSV la base de datos. | | |
| Duración: 4 días | Esfuerzo: 3 días | Necesario para historias de usuario: 1 |

Tabla 14. Tarea 5: Cargar CSV a BD para la web

| | | |
|--|--|--|
| Nombre del integrante: Alberto Pons Buttazzo | | |
| Descripción de la actividad: implementar funcionalidad para cargar los registros de la base de datos a una tabla. | | |

| | | |
|-------------------------|------------------------|--|
| Duración: 2 días | Esfuerzo: 1 día | Necesario para historias de usuario: 2 |
|-------------------------|------------------------|--|

Tabla 15. Tarea 6: Cargar registros a tabla para la web

| | | |
|---|---------------------------|--|
| Nombre del integrante: Alberto Pons Buttazzo | | |
| Descripción de la actividad: implementar funcionalidades extras sobre la tabla (filtros, eliminaciones y búsquedas). | | |
| Duración: 1 semana y 1 día | Esfuerzo: 1 semana | Necesario para historias de usuario: 2 |

Tabla 16. Tarea 7: Funcionalidades extras sobre la tabla para la web

| | | |
|--|-------------------------|--|
| Nombre del integrante: Alberto Pons Buttazzo | | |
| Descripción de la actividad: implementar diseño sobre la ventana donde se muestra la tabla. | | |
| Duración: 4 días | Esfuerzo: 3 días | Necesario para historias de usuario: 2 |

Tabla 17. Tarea 8: Diseño sobre ventana de la tabla para la web

| | | |
|---|----------------------------|--|
| Nombre del integrante: Alberto Pons Buttazzo | | |
| Descripción de la actividad: implementar funcionalidades para generar las vistas (gráficos). Como por ejemplo, calcular el número de registros totales, los pesos medios por fecha o la variación de los pesos en el tiempo. | | |
| Duración: 2 semanas | Esfuerzo: 2 semanas | Necesario para historias de usuario: 3 |

Tabla 18. Tarea 9: Funcionalidades de las vistas para la web

| | | |
|---|-------------------------|--|
| Nombre del integrante: Alberto Pons Buttazzo | | |
| Descripción de la actividad: implementar diseño sobre la ventana donde se muestran las vistas. | | |
| Duración: 4 días | Esfuerzo: 3 días | Necesario para historias de usuario: 3 |

Tabla 19. Tarea 10: Diseño sobre ventana de las vistas para la web

| | | |
|--|-------------------------|--|
| Nombre del integrante: Alberto Pons Buttazzo | | |
| Descripción de la actividad: implementar funcionalidades extras que haga en falta el cliente. Como por ejemplo, generar más funcionalidades para la tabla o las vistas, o implementar criterios de internalización o accesibilidad. | | |
| Duración: 6 días | Esfuerzo: 5 días | Necesario para historias de usuario: 1, 2, y/o 3 |

Tabla 20. Tarea 11: Funcionalidades extras restantes para la web

| | | |
|--|-------------------------|---|
| Nombre del integrante: Alberto Pons Buttazzo | | |
| Descripción de la actividad: implementar diseño sobre las nuevas funcionalidades solicitadas o implementar cambios en funcionalidades anteriores. | | |
| Duración: 4 días | Esfuerzo: 3 días | Necesario para historias de usuario: 1, 2 y/o 3 |

Tabla 21. Tarea 12: Finalización de diseño para la web

Antes de continuar definiendo las tareas de las historias de usuario 4 y 5, se va a recapitular el esfuerzo y duración estimados para las historias de usuario 1, 2 y 3 en la tabla 22 (se redondean los valores). También es importante matizar la diferencia entre esfuerzo y duración. Se entiende por esfuerzo el tiempo directamente invertido por una persona en la tarea (tiempo ideal), y duración, como el tiempo total en completar la tarea. Este último incluye dependencias, revisiones y autorizaciones externas.

| | Base + Historia de usuario 1: Importación CSV | Historia de usuario 2: Visualización de datos | Extras finales + Historia de usuario 3: Generación de vistas | Total |
|-----------------|--|--|---|--------------|
| Esfuerzo | 8 días | 16 días | 22 días | 46 días |
| Duración | 11 días | 20 días | 25 días | 56 días |

Tabla 22. Esfuerzo y duración estimados en las historias de usuario 1, 2 y 3

Se continua con la definición de las tareas de las historias de usuario 4 y 5:

| | | |
|---|------------------------|--|
| Nombre del integrante: Alberto Pons Buttazzo | | |
| Descripción de la actividad: preparar el entorno de implementación para empezar a desarrollar la aplicación inteligente. Adquirir y configurar herramientas que se van a usar (Jupyter Notebook y Google Colab). | | |
| Duración: 2 día | Esfuerzo: 1 día | Necesario para historias de usuario: 4 y 5 |

Tabla 23. Tarea 13: Entorno de implementación para la app inteligente

| | | |
|---|-------------------------|--|
| Nombre del integrante: Alberto Pons Buttazzo | | |
| Descripción de la actividad: investigar y empezar a probar distintas alternativas, librerías y algoritmos de Machine Learning (Arima y Prophet). | | |
| Duración: 6 días | Esfuerzo: 5 días | Necesario para historias de usuario: 4 y 5 |

Tabla 24. Tarea 14: Investigación inicial de Machine Learning con Arima y Prophet para la app inteligente

| | | |
|---|-------------------------|--|
| Nombre del integrante: Alberto Pons Buttazzo | | |
| Descripción de la actividad: usar el modelo Prophet para predecir la variación de los pesos con el tiempo y generar las primeras métricas para evaluar la precisión del algoritmo. | | |
| Duración: 6 días | Esfuerzo: 5 días | Necesario para historias de usuario: 4 y 5 |

Tabla 25. Tarea 15: Predicción de la variación de los pesos con Prophet para la app inteligente

| | | |
|--|--|--|
| Nombre del integrante: Alberto Pons Buttazzo | | |
| Descripción de la actividad: usar el modelo Arima para predecir la variación de los pesos con el tiempo y generar las métricas para evaluar la precisión del algoritmo. | | |

| | | |
|---------------------------|-------------------------|--|
| Duración: 1 semana | Esfuerzo: 5 días | Necesario para historias de usuario: 4 y 5 |
|---------------------------|-------------------------|--|

Tabla 26. Tarea 16: Predicción de la variación de los pesos con Arima para la app inteligente

| | | |
|---|-------------------------|--|
| Nombre del integrante: Alberto Pons Buttazzo | | |
| Descripción de la actividad: generar con Prophet las métricas finales para terminar con la optimización del algoritmo. | | |
| Duración: 1 semana | Esfuerzo: 5 días | Necesario para historias de usuario: 5 |

Tabla 27. Tarea 17: Predicción final de la variación de los pesos con Prophet para la app inteligente

Antes de continuar definiendo las tareas de la historia de usuario 6, se va a recapitular el esfuerzo y duración estimados para las historias de usuario 4 y 5 en la tabla 28.

| | Historia de usuario 4: Implementación de algoritmos | Historia de usuario 5: Generación de predicciones | Total |
|-----------------|--|--|--------------|
| Esfuerzo | 8 días | 11 días | 19 días |
| Duración | 11 días | 17 días | 28 días |

Tabla 28. Esfuerzo y duración estimados en las historias de usuario 4 y 5

Se continua con la definición de las tareas de las historia de usuario 6:

| | | |
|--|------------------------|--|
| Nombre del integrante: Alberto Pons Buttazzo | | |
| Descripción de la actividad: preparar el entorno de implementación para empezar a desarrollar la integración de la aplicación inteligente con la aplicación web. Adquirir y configurar herramientas que se van a usar (Anaconda Navigator). | | |
| Duración: 2 días | Esfuerzo: 1 día | Necesario para historias de usuario: 6 |

Tabla 29. Tarea 18: Entorno de implementación para la integración

| | | |
|---|-------------------------|--|
| Nombre del integrante: Alberto Pons Buttazzo | | |
| Descripción de la actividad: conectar la aplicación inteligente con la aplicación web. | | |
| Duración: 5 días | Esfuerzo: 3 días | Necesario para historias de usuario: 6 |

Tabla 30. Tarea 19: Conexión de ambas aplicaciones para la integración

| | | |
|---|------------------------|--|
| Nombre del integrante: Alberto Pons Buttazzo | | |
| Descripción de la actividad: generar datos de entrada que la aplicación web debe enviar a la aplicación inteligente. | | |
| Duración: 2 días | Esfuerzo: 1 día | Necesario para historias de usuario: 6 |

Tabla 31. Tarea 20: Datos de entrada para la integración

| | | |
|--|------------------------|--|
| Nombre del integrante: Alberto Pons Buttazzo | | |
| Descripción de la actividad: modificar aplicación inteligente para poder recibir los datos de la aplicación web y poder enviar los resultados predichos.. | | |
| Duración: 2 días | Esfuerzo: 1 día | Necesario para historias de usuario: 6 |

Tabla 32. Tarea 21: Modificación de la app inteligente para la integración

| | | |
|--|------------------------|--|
| Nombre del integrante: Alberto Pons Buttazzo | | |
| Descripción de la actividad: generar datos de salida que la aplicación inteligente debe enviar a la aplicación web. | | |
| Duración: 1 días | Esfuerzo: 1 día | Necesario para historias de usuario: 6 |

Tabla 33. Tarea 22: Datos de salida para la integración

| | | |
|---|--|--|
| Nombre del integrante: Alberto Pons Buttazzo | | |
| Descripción de la actividad: recibir datos generados por la aplicación inteligente y generar vistas para mostrarlos. | | |

| | | |
|-------------------------|------------------------|--|
| Duración: 2 días | Esfuerzo: 1 día | Necesario para historias de usuario: 6 |
|-------------------------|------------------------|--|

Tabla 34. Tarea 23: Generación de vistas para la integración

Finalmente se recapitulan el esfuerzo y duración estimados para la historia de usuario 6 en la tabla 35.

| | Historia de usuario 6: Integración de ambas aplicaciones | Total |
|-----------------|---|--------------|
| Esfuerzo | 8 días | 8 días |
| Duración | 14 días | 14 días |

Tabla 35. Esfuerzo y duración estimados en las historia de usuario 6

5.2. PLANIFICACIÓN TEMPORAL DEL PROYECTO

5.2.1. DEFINICIÓN DE LOS SPRINTS

En base a las duraciones obtenidas, se ha decidido dividir el proyecto en 7 sprints, cada uno de ellos con una duración de 2 semanas. Se decide elegir este tipo de duración de los sprints en base a los siguientes criterios (NeuronForest, 2020):

- Favorece sistemas recién empezados.
- Beneficia a equipos de desarrollo noveles.
- Facilita y agiliza el reajuste.
- Maximiza el acierto en la anticipación (planificaciones más cortas).
- Permite detectar retrasos de forma más rápida.
- Se consigue un mayor feedback al tener un contacto más continuo con el cliente (reuniones más asiduas).

Finalmente, se obtiene un total de 3 meses y medio de desarrollo, lo que resulta en el siguiente reparto en sprints representado en la tabla 36:

| Sprint | Asignación |
|---------------|---|
| Sprint 1 | Base + Historia de usuario 1: Importación de CSV |
| Sprint 2 | Historia de usuario 2: Visualización de datos |
| Sprint 3 | 1/2 Historia de usuario 3: Generación de vistas |
| Sprint 4 | Extras finales + 2/2 Historia de usuario 3: Generación de vistas |
| Sprint 5 | Historia de usuario 4: Implementación de algoritmos |
| Sprint 6 | Historia de usuario 5: Generación de predicciones |
| Sprint 7 | Historia de usuario 6: Integración de ambas aplicaciones |

Tabla 36. Asignación de los sprints

5.2.2. DIAGRAMA TEMPORAL DE SPRINTS

En base a los sprints detectados, se consigue el siguiente diagrama de la ilustración 5:

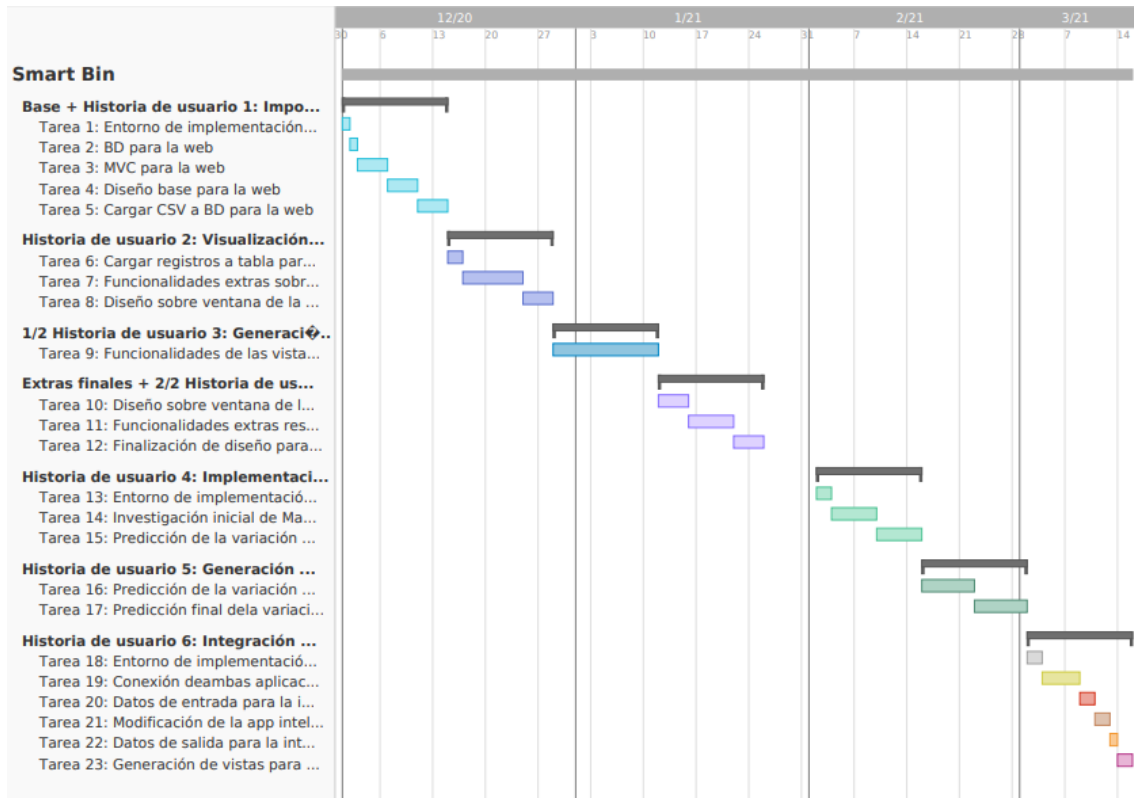


Ilustración 5. Diagrama de Gantt Inicial

Cabe destacar, que tanto la definición de tareas, como la planificación de las mismas se separan temporalmente en 3 etapas:

- La primera definición y planificación inicial ocurren el 01/12/2020 y conciernen a los 4 primeros Sprints, aquellos que se corresponden con la aplicación web.
- La segunda definición y planificación inicial ocurren el 02/02/2021 y conciernen a los 2 siguientes Sprints, aquellos que se corresponden con la aplicación inteligente.
- La última definición y planificación inicial ocurren el 02/03/2021 y conciernen al último Sprint, aquel que se corresponde con la integración de ambas aplicaciones.

A diferencia de una agilidad pura, donde cada planificación se plantea al inicio de un Sprint, en este caso se plantea una planificación al inicio de cada épica (nivel más alto que una historia de usuario), es decir, para el desarrollo de la aplicación web, la aplicación inteligente y para la integración de ambas aplicaciones.

5.3. VALORACIÓN DE LA DEDICACIÓN Y EL COSTE ECONÓMICO

5.3.1. VALORACIÓN ESTIMADA DE LA DEDICACIÓN Y EL COSTE ECONÓMICO

Para valorar el coste económico total del proyecto se va a tener en cuenta los recursos hardware y software necesarios, el salario, y otros costes, como la luz o internet:

- Costes materiales:
 - Costes de internet: $50 \text{ (euros al mes)} * 3,5 \text{ (meses de duración del proyecto)} * 1/3 \text{ (porcentaje dedicado al proyecto)} = 58 \text{ €}$.
 - Costes de recursos software (antivirus): $6 \text{ (euros al mes)} * 3,5 \text{ (meses de duración del proyecto)} * 1/3 \text{ (porcentaje dedicado al proyecto)} = 7 \text{ €}$.
 - Costes de recursos hardware (ordenador): $850 \text{ (precio total en euros)} / 60 \text{ (meses de vida útil)} * 3,5 \text{ meses (duración del proyecto)} * 1/2 \text{ (porcentaje dedicado al proyecto)} = 25 \text{ €}$.
- Costes de recursos humanos:
 - Salario del desarrollador: $19000 \text{ (salario medio bruto de programador al año en euros) (Hack a Boss, 2019)} / 12 \text{ (al mes)} / 176 \text{ (horas al mes)} * 490 \text{ (horas de trabajo totales, 5 horas por día durante 98 días)} = 4408 \text{ €}$.
 - Salario del Scrum Master: $42.000 \text{ (salario medio bruto de Scrum Master al año en euros) (glassdoor, 2021)} / 12 \text{ (al mes)} / 176 \text{ (horas al mes)} * 7 \text{ (1 hora de trabajo por Sprint)} = 140 \text{ €}$.
- Costes indirectos (materiales, luz, desplazamiento u otros servicios no atribuibles directamente a la producción del software) (BidDown, 2020): $3001 \text{ euros (total de gastos)} + 3001 \text{ euros} * 0,2 \text{ (+20\%)} = 5458 \text{ €}$.

Finalmente a esos 5458 € se le debe restar un 15% de IRPF como autónomo, obteniendo un coste final de **4639 €** para la totalidad del proyecto.

Todos los costes quedan reflejados en la tabla 37.

| | Costes (€) |
|---|-------------------|
| Costes de internet | 58 |
| Costes de recursos SW | 7 |
| Costes de recursos HW | 25 |
| Salarios | 4548 |
| Total con gastos indirectos (+20%) | 5458 |
| TOTAL (-15%) | 4639 |

Tabla 37. Costes estimados del proyecto

5.3.2. VALORACIÓN REAL DE LA DEDICACIÓN Y EL COSTE ECONÓMICO

Tras finalizar el desarrollo, resulta que se han dedicado al proyecto un total de 390 horas de trabajo. Teniendo en cuenta que en cada día se han establecido 5 horas de trabajo de media, el tiempo en días se traduce a 78. Si hay 22 días laborables al mes, el resultado en meses resulta en 3.55 meses de trabajo. En la planificación inicial se planificaron 3,5 meses y 490 horas de trabajo. Si se establecen esos parámetros sobre las operaciones de costes que se realizaron en el punto anterior, resulta la siguiente tabla 38:

| | Costes (€) |
|---|-------------------|
| Costes de internet | 59 |
| Costes de recursos SW | 7 |
| Costes de recursos HW | 25 |
| Salarios | 3668 |
| Total con gastos indirectos (+20%) | 4511 |
| TOTAL (-15%) | 3834 |

Tabla 38. Costes reales del proyecto

Al final se obtiene una diferencia de 850 € a devolver, esto es entendible, ya que por lo general cada Sprint se terminaba antes, con lo que al final se restan varios días sin trabajar, más concretamente, 20 días.

6. DESARROLLO DEL CONTENIDO DEL PROYECTO

6.1. ESTRUCTURA BÁSICA DE DESARROLLO

Conforme a la metodología elegida en la sección 3 (Scrum), se va a organizar el desarrollo en 7 Sprints. Cada Sprint dura dos semanas y se divide a su vez en varias reuniones (Schwaber & Sutherland, 2013):

- Reunión inicial para planificar el proyecto (Sprint Planning Meeting). Se seleccionan el subconjunto de tareas del Product Backlog que se van a ejecutar durante el Sprint y la metodología que se va a seguir para conseguirlo.(Sprint Backlog), al igual que se establece el objetivo que se quiere conseguir (Sprint Goal).
 - Tiempo esperado: 1 hora.
 - Resultados: Sprint Backlog y Sprint Goal.
 - Participantes: Desarrollador, Product Owner y Scrum Master.
- Reuniones diarias (Daily Scrum), que al haber únicamente una persona en el equipo de desarrollo, resulta en un tiempo previo para autoorganizar el trabajo que se ejecutará durante el día.
 - Tiempo esperado: 15 minutos.
 - Participante: Desarrollador.
- Reunión que se realiza al final del Sprint para revisar (Sprint Review) los elementos del Sprint Backlog que se han completado (Incremento) y medir el progreso del proyecto (Burdown).
 - Tiempo esperado: 30 minutos.
 - Resultado: Incremento y Burdown.
 - Participantes: Desarrollador, Product Owner y Scrum Master.

- Reunión para analizar la retrospectiva (Sprint Retrospective), que permite estudiar el enfoque inicial y final del Sprint para identificar posibles mejoras para futuros Sprints.
 - Tiempo esperado: 30 minutos.
 - Participantes: Desarrollador, Product Owner y Scrum Master.

6.2. SPRINT 1 - BASE + HISTORIA DE USUARIO 1: IMPORTACIÓN DE CSV

6.2.1. SPRINT 1 PLANNING MEETING

Durante la primera reunión de planificación se establecen el Sprint Backlog y el Sprint Goal para el primer Sprint en la tabla 39 e ilustración 6.

| SPRINT BACKLOG | |
|---|--|
| Tareas | Metodologías |
| Tarea 1: Entorno de implementación para la web. | Metodología 1: Descargar y configurar las herramientas XAMMP y Visual Studio Code, e incorporar plugins de apoyo. |
| Tarea 2: BD para la web. | Metodología 2: Crear la tabla para almacenar los registros en phpMyAdmin, siguiendo las columnas generadas por el CSV del sensor. |
| Tarea 3: MVC para la web. | Metodología 3: Generar estructura base de la web, siguiendo el patrón Modelo-Vista-Controlador. |
| Tarea 4: Diseño base para la web. | Metodología 4: Implementar diseño base de la aplicación web (cabecera, cuerpo y pie de página), incorporando las etiquetas necesarias (head, body y footer). |

| | |
|---|--|
| Tarea 5: Cargar CSV a BD para la web. | Metodología 5: Crear función para importar los datos del CSV a la base de datos. |
| SPRINT GOAL | |
| El objetivo del Sprint 1 es el de preparar la base de la aplicación web y permitir al usuario que pueda importar sus registros en formato CSV a la base de datos de la web. | |

Tabla 39. Sprint 1 Backlog

| Estado | Fecha Inicio | Fecha Fin | Esfuerzo planificado | Duración planificada | |
|--|--------------|---------------|----------------------|----------------------|---------|
| ● Sprint 1: Base + Historia de usuario 1: Importación de CSV | | | | | |
| Tarea 1: Entorno de implementación para la web | En espera | dic. 1, 2020 | dic. 1, 2020 | 4 hours | 5hours |
| Tarea 2: BD para la web | En espera | dic. 2, 2020 | dic. 2, 2020 | 4 hours | 5hours |
| Tarea 3: MVC para la web | En espera | dic. 3, 2020 | dic. 6, 2020 | 16 hours | 20hours |
| Tarea 4: Diseño base para la web | En espera | dic. 7, 2020 | dic. 10, 2020 | 16 hours | 20hours |
| Tarea 5: Cargar CSV a BD para la web | En espera | dic. 11, 2020 | dic. 14, 2020 | 16 hours | 20hours |
| + Agregar Activity | | | | | |
| | | | dic. 1 - 11 | dic. 1 - 14 | |
| | | | 56 hours | 70hours | |
| | | | Total | Total | |

Ilustración 6. Sprint 1 - Sprint Backlog inicial

6.2.2. SPRINT 1 ANÁLISIS MEDIANTE CASOS DE USO

El siguiente diagrama de casos de uso de la ilustración 7, muestra la única funcionalidad que se va a implementar en la web en el Sprint 1 (importar el CSV con los datos del sensor):

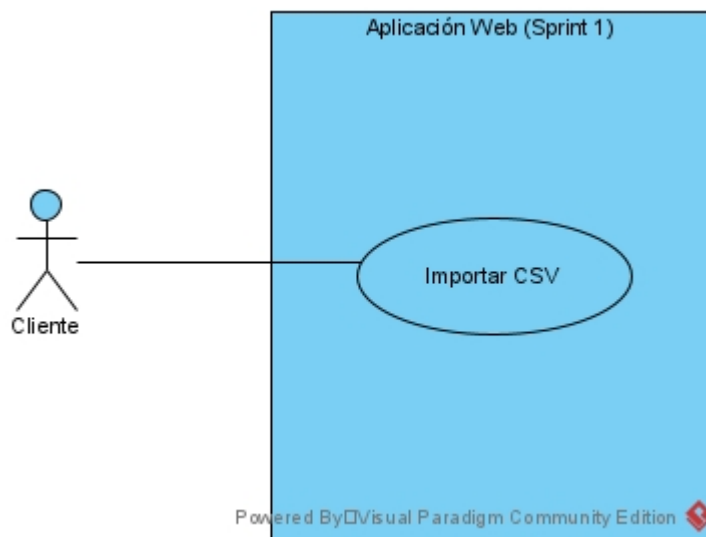


Ilustración 7. Diagrama casos de uso del Sprint 1

6.2.3. SPRINT 1 DISEÑO MEDIANTE UML

El siguiente diagrama UML de la ilustración 8, muestra la estructura base de la web que sigue el patrón Modelo-Vista-Controlador:

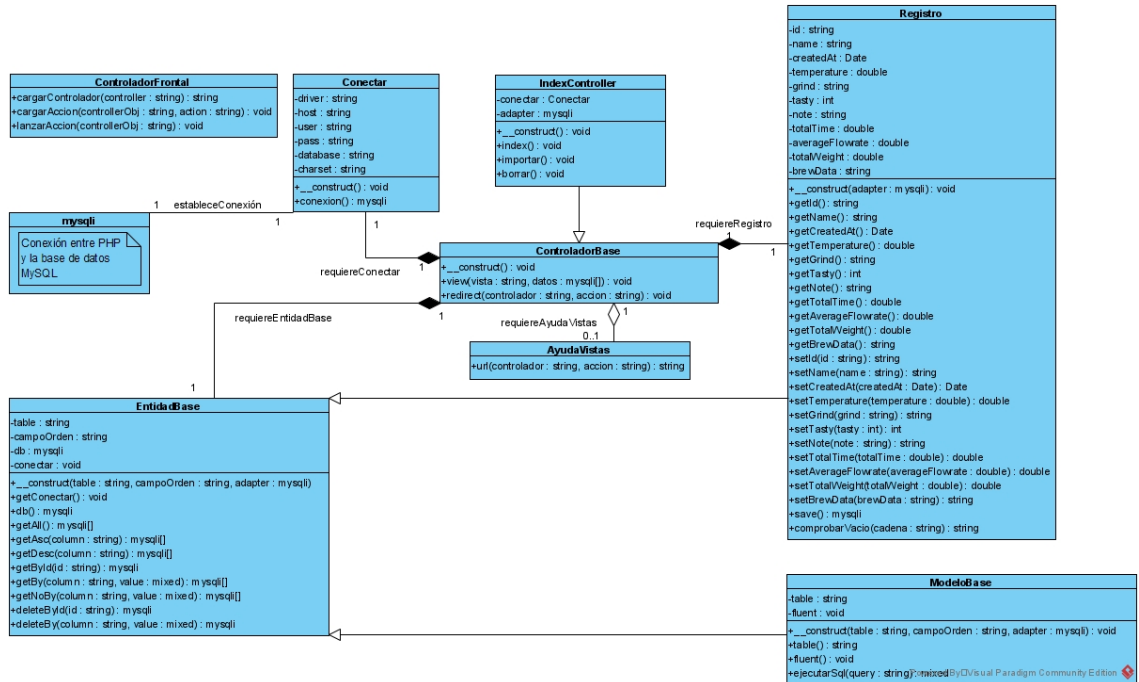


Ilustración 8. Diagrama UML del Sprint 1

6.2.4. SPRINT 1 DESARROLLO

En primer lugar, se establece una organización sobre la tarea 1 (*Entorno de implementación para la web*) en el primer Daily Scrum del Sprint:

- Descargar y configurar XAMPP (Apache Friends, 2020).
- Descargar y configurar Visual Studio Code (Microsoft, 2020).

Cabe destacar algunas consideraciones de configuración para ambos sistemas. En el caso de XAMPP, es necesario modificar el puerto del servidor Apache al entrar en conflicto con un puerto ya usado (se modifican los ficheros *httpd.conf* y *http-ssl.conf*) (Subit, 2019). Por otro lado, en Visual Studio Code se instalan los siguientes plugins:

- CSS Peek para localizar las definiciones CSS (Prakash, 2020).
- Live Server para recargar en vivo la web (Dey, 2019).

- Total weight (double): peso máximo durante el registro.
- Brew data (cadena): cadena con todos los pesos del registro separados por ";".
- Otros: temperature (double), grind (cadena) y tasty (entero).

A continuación, se planifica la tarea 3 (*MVC para la web*) (Robles, 2021):

- Crear el modelo para los registros.
- Generar la conexión y las operaciones con la base de datos.
- Generar los controladores de las vistas.
- Implementar el índice principal.

Se comienza creando el modelo *Registro*, que contiene todos los atributos de los registros del sensor, recuperaciones y asignaciones de los mismos (*getters* y *setters*), y otros métodos para añadir y buscar. Además, dispone de otros muchos métodos al extender de la clase *EntidadBase*, que se explica a continuación.

Posteriormente, se genera la conexión con la base de datos phpMyAdmin en la clase *Conectar*, siguiendo la configuración establecida en el fichero de configuración *database*. También se crean una serie de métodos que reflejan distintas transacciones con la base de datos en la clase *EntidadBase* (eliminaciones y selecciones).

Seguidamente, se crean los distintos controladores base para las vistas de la web. Es necesaria una clase para cargar las vistas (*ControladorBase*) y un fichero (*ControladorFrontal*) con funciones que carguen los controladores (intermediarios entre el modelo y las vistas) y las acciones (métodos de los controladores).

Finalmente, se genera el índice principal de la página (*index*) donde convergen todas las vistas (se cargan controladores y acciones).

Una vez creado el MVC, se organiza la tarea 4 (*Diseño base para la web*):

- Crear la cabecera, cuerpo y pie de página de la web.

- Implementar primer diseño de la web.

Se comienza con una estructuración básica para la web que incluye:

- Una cabecera (*headerBase1* y *headerBase2*) con metadatos, enlaces a hojas de estilo y scripts, y una barra de navegación.
- Un cuerpo vacío.
- Un pie de página con la información de copyright y todos los enlaces de los scripts que se usan.

Además, se crea una hoja de estilo propia (*main*) y se decide hacer uso de Bootstrap (Bootstrap, 2021), biblioteca multiplataforma para el diseño web. Cabe destacar, que se decide usar una base de color azulado, siguiendo algunos esquemas de color de la página *Hook Agency* (Brown, 2020).

Finalmente, para acabar con el desarrollo del primer Sprint, se planifica la última tarea del mismo (*Cargar CSV a BD para la web*):

- Implementar el método para importar los datos del CSV a la base de datos de phpMyAdmin.

Se genera un formulario que envía el CSV al controlador del índice (phppot, 2020). Una vez allí, se recorre cada fila y se va insertando en la base de datos con el método *importar* (que a su vez llama al modelo para ejecutar las inserciones).

6.2.5. SPRINT 1 REVIEW

Durante la reunión de revisión (Sprint Review), en primer lugar se revisan los elementos del Sprint Backlog de la ilustración 10 que se han completado durante el desarrollo del primer Sprint (Incremento):

| Incremento de Sprint 1: Base + Historia de usuario 1: Importaci... | Estado | Fecha Inicio | Fecha Fin | Esfuerzo real | Duración real |
|--|--------|--------------|---------------|-------------------|------------------|
| Tarea 1: Entorno de implementación para la web | Listo | dic. 6, 2020 | dic. 7, 2020 | 4 hours | 5hours |
| Tarea 2: BD para la web | Listo | dic. 7, 2020 | dic. 8, 2020 | 4 hours | 5hours |
| Tarea 3: MVC para la web | Listo | dic. 7, 2020 | dic. 8, 2020 | 8 hours | 10hours |
| Tarea 4: Diseño base para la web | Listo | dic. 8, 2020 | dic. 12, 2020 | 8 hours | 10hours |
| Tarea 5: Cargar CSV a BD para la web | Listo | dic. 8, 2020 | dic. 12, 2020 | 16 hours | 20hours |
| + Agregar Activity | | | | | |
| | | | | 40 hours Total | 50hours Total |

Ilustración 10. Sprint 1 - Incremento

Seguidamente la reunión finaliza tras medir el progreso del proyecto mediante la gráfica de la ilustración 11, que refleja el progreso del trabajo remanente ideal y real en días (Burdown):

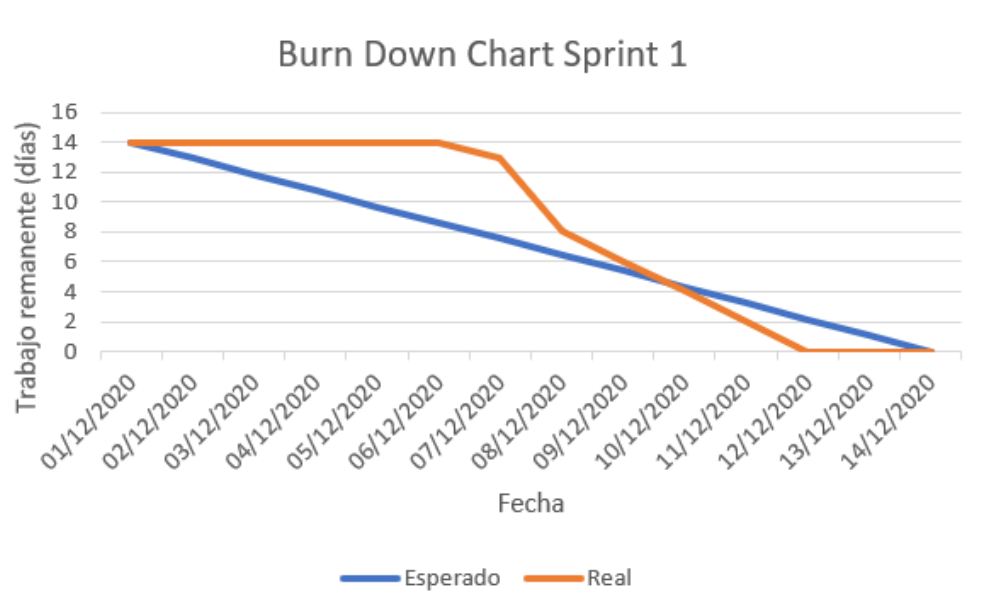


Ilustración 11. Burn Down Chart Sprint 1

6.2.6. SPRINT 1 RETROSPECTIVE

En base al Incremento y Burndown reflejados en el punto anterior, se estiman el enfoque inicial y final del Sprint, y las posibles mejoras para futuros Sprints.

El enfoque inicial marcaba un desarrollo excesivamente progresivo a largo de los días. En cambio, el enfoque final demuestra una primera semana sin prácticamente progreso al usar la mayor parte del tiempo en iniciar la investigación. Seguidamente se experimenta un cúmulo de trabajo entre la primera y segunda semana, que va reduciéndose hasta el día 12/12/2020, finalizando así el Sprint 1 (en el enfoque inicial se estimó que finalizaría el

14/12/2020). Además, las horas de trabajo se ven reducidas tras una investigación previa y reutilización de componentes usados en anteriores trabajos (principalmente para la creación del MVC). La principal y más evidente mejora sugiere un mayor paralelismo entre la investigación y el desarrollo, promoviendo así, un trabajo con mayor progresividad.

6.3. SPRINT 2 - HISTORIA DE USUARIO 2: VISUALIZACIÓN DE DATOS

6.3.1. SPRINT 2 PLANNING MEETING

Durante la reunión de planificación se establecen el Sprint Backlog y el Sprint Goal para el segundo Sprint en la tabla 40 e ilustración 12.

| SPRINT BACKLOG | |
|---|---|
| Tareas | Metodologías |
| Tarea 6: Cargar registros a tabla para la web. | Metodología 6: Recuperar los registros de la base de datos y cargarlos en una tabla. |
| Tarea 7: Funcionalidades extras sobre la tabla para la web. | Metodología 7: Crear funciones extras sobre la tabla (filtros, eliminaciones y búsquedas), haciendo uso de sentencias SQL a la base de datos. |
| Tarea 8: Diseño sobre ventana de la tabla para la web. | Metodología 8: Implementar diseño sobre la ventana donde se muestra la tabla, haciendo uso de Bootstrap y de scripts que ofrezcan dinamismo a la tabla. |
| SPRINT GOAL | |
| El objetivo del Sprint 2 es el de generar una primera visualización de los registros a través de una tabla y permitir al usuario que pueda interactuar de distintas maneras con la misma. | |

Tabla 40. Sprint 2 Backlog

| Actividad | Estado | Fecha Inicio | Fecha Fin | Esfuerzo planificado | Duración planificada |
|--|-----------|---------------|---------------|--------------------------|-------------------------|
| Tarea 6: Cargar registros a tabla para la web | En espera | dic. 15, 2020 | dic. 16, 2020 | 8 hours | 10hours |
| Tarea 7: Funcionalidades extras sobre la tabla para la web | En espera | dic. 17, 2020 | dic. 24, 2020 | 32 hours | 40hours |
| Tarea 8: Diseño sobre ventana de la tabla para la web | En espera | dic. 25, 2020 | dic. 28, 2020 | 16 hours | 20hours |
| | | | | 56 hours Total | 70hours Total |

Ilustración 12. Sprint 2 - Sprint Backlog inicial

6.3.2. SPRINT 2 ANÁLISIS MEDIANTE CASOS DE USO

El siguiente diagrama de casos de uso de la ilustración 13, muestra las funcionalidades que se van a implementar en la web en el Sprint 2:

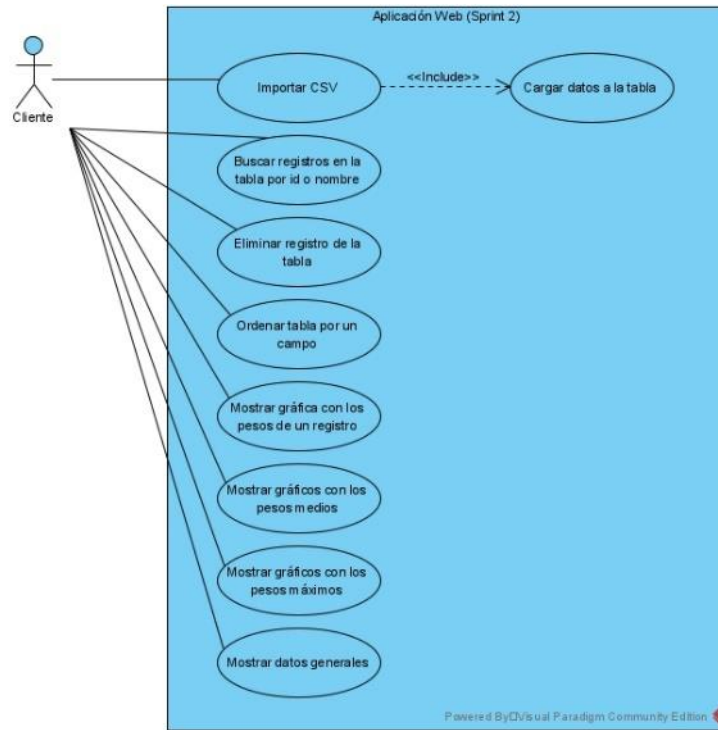


Ilustración 13. Diagrama casos de uso del Sprint 2

6.3.3. SPRINT 2 DISEÑO MEDIANTE UML

El siguiente diagrama UML de la Ilustración 14, muestra la estructura Modelo-Vista-Controlador, junto a las funcionalidades implementadas en el Sprint 2 (se incluye la clase *VistasController* y varios métodos):

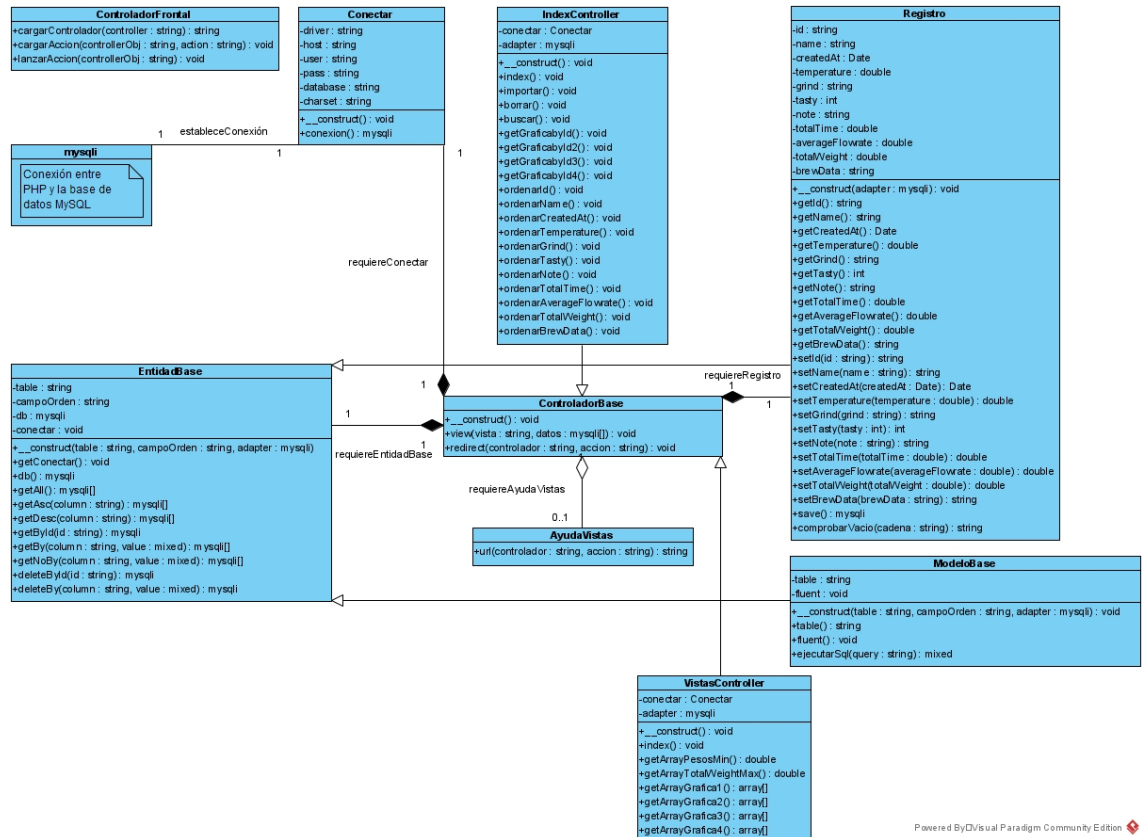


Ilustración 14. Diagrama UML del Sprint 2

6.3.4. SPRINT 2 DESARROLLO

En primer lugar, se establece una organización sobre la tarea 6 (*Cargar registros a tabla para la web*) en el primer Daily Scrum del Sprint:

- Crear una tabla con las mismas columnas que el registro.
- Recorrerla con las filas de la tabla *registro*.

Se parte de la ventana que se creó en el anterior Sprint, asignándole el nombre de *Inicio* y creando su controlador *IndexController*. Seguidamente, se crea una tabla estándar en HTML, con las mismas columnas que tiene cada registro. Posteriormente, se crea una funcionalidad para obtener todos los registros de la tabla *registro* en la extensión del modelo (*EntidadBase*), para que puedan ser cargados por el controlador a la vista *Inicio*.

En segundo lugar, se organiza la tarea 7 (*Funcionalidades extras sobre la tabla para la web*):

- Buscar registros por identificador o nombre.

- Eliminar registros.
- Ordenar registros por columna de forma ascendente.

Similarmente a la tarea anterior, se crean las funcionalidades de búsqueda, eliminación y ordenación ascendente en el modelo y/o su extensión. Posteriormente en la ventana de *Inicio*, se crean el formulario de búsqueda, un botón de eliminación en cada fila de la tabla, y un botón de ordenación en cada columna de la tabla. Cada uno de ellos llama al controlador y este carga los resultados a la vista *Inicio* (también se muestran mensajes de alerta para la búsqueda).

A continuación, se planifica la tarea 8 (*Diseño sobre ventana de la tabla para la web*):

- Implementar diseño a los formularios y a la tabla.

Se incluyen nuevas características de diseño en la hoja de estilo *main*. Asegurando que el texto en la tabla mantenga una correcta alineación, y que los botones de los formularios y la tabla tengan el color azulado de la web.

Finalmente, debido a una finalización temprana de las tareas antes del vencimiento del Sprint 2, se decide adelantar trabajo de la tarea 9 (*Funcionalidades de las vistas para la web*):

- Implementar datos generales, y gráficos sobre los pesos máximos y medios.
- Implementar gráfico sobre la variación de los pesos en el tiempo y que sea interactivo con la tabla.

Se crea una nueva ventana llamada *Vistas* y su controlador *VistasController*. Seguidamente, se implementan en el modelo funcionalidades para calcular una serie de datos generales (número de registros o peso mínimo que no sea 0) y otras para calcular los valores de los ejes X e Y de las gráficas que se van a crear (sobre los pesos máximos y medios). Estos datos los recibe el controlador *VistasController*, para que puedan ser cargados a la vista *Vistas*.

Asimismo, se añade una de las funcionalidades más importantes de toda la web, que es el cálculo de los ejes X e Y para la variación de los pesos por

instante de tiempo. Se incluyen 4 métodos (sólo se implementa el primero) que difieren en el eje Y (siendo X siempre el instante de tiempo):

- Primer método *getGraficabyId*:
 - Eje Y: peso en gramos.
- Segundo método *getGraficabyId2*:
 - Eje Y: peso diferente no seguido en gramos.
- Tercer método *getGraficabyId3*:
 - Eje Y: peso diferente no seguido en su parte entera en gramos.
- Cuarto método *getGraficabyId4*:
 - Eje Y: peso medio en distintas franjas en gramos.

Por último, son necesarios scripts para poder visualizar dichos gráficos, siendo de gran referencia la página *canvasJS* (Fenopix, 2021).

6.3.5. SPRINT 2 REVIEW

Durante la reunión de revisión, en primer lugar se revisan los elementos del Sprint Backlog de la ilustración 15 que se han completado durante el desarrollo del segundo Sprint:

| Incremento de Sprint 2: Historia de usuario 2: Visualización de ... | Estado | Fecha Inicio | Fecha Fin | Esfuerzo planificado | Duración real | |
|---|-------------------|---------------|---------------|----------------------|------------------|--|
| Tarea 6: Cargar registros a tabla para la web | Listo | dic. 15, 2020 | dic. 15, 2020 | 4 hours | 5hours | |
| Tarea 7: Funcionalidades extras sobre la tabla para la web | Listo | dic. 16, 2020 | dic. 27, 2020 | 24 hours | 30hours | |
| Tarea 8: Diseño sobre ventana de la tabla para la web | Listo | dic. 16, 2020 | dic. 27, 2020 | 12 hours | 15hours | |
| Tarea 9: Funcionalidades de las vistas para la web | Trabajando en ... | dic. 19, 2020 | | | | |
| + Agregar Activity | | | | | | |
| | | | | 40 hours Total | 50hours Total | |

Ilustración 15. Sprint 2 - Incremento

Seguidamente la reunión finaliza tras medir el progreso del proyecto mediante la gráfica de la ilustración 16:

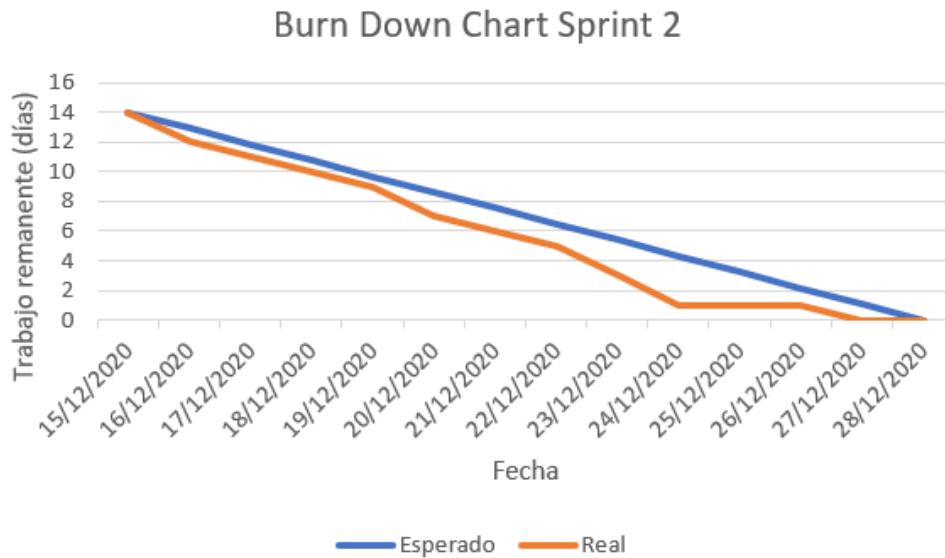


Ilustración 16. Burn Down Chart Sprint 2

6.3.6. SPRINT 2 RETROSPECTIVE

El enfoque inicial marcaba un desarrollo progresivo a lo largo de los días. Asimismo, el enfoque final demuestra que se cumple dicho desarrollo, incluso se acaba antes de los esperado el día 27/12/2020 (en el enfoque inicial se estimó que finalizaría el 28/12/2020). Cómo se estimó oportuno en la reunión de retrospectiva del primer Sprint, se cumplieron las expectativas manteniendo la progresividad deseada. Las posibles mejoras en futuros Sprints sugieren un reparto de tareas más próximo a la realidad, ya que en este Sprint se tuvo que adelantar trabajo del siguiente Sprint al acabar antes de lo previsto. Aunque las dos últimas tareas reflejen una fecha de finalización el 27/12/2020, la mayor parte del trabajo fue finalizado el 24/12/2020.

6.4. SPRINT 3 - 1/2 HISTORIA DE USUARIO 3: GENERACIÓN DE VISTAS

6.4.1. SPRINT 3 PLANNING MEETING

Durante la reunión de planificación se establecen el Sprint Backlog y el Sprint Goal para el tercer Sprint en la tabla 41 e ilustración 17. En este caso, debido a que la tarea 9 se empezó a desarrollar en el anterior Sprint, su metodología va a ser modificada levemente. Además, se demandan ciertos cambios visuales y de medidas, para la tabla y las vistas. Surge entonces una

nueva tarea en la tabla 42, que se identifica con el nombre: *Tarea 9.1: Cambios en vistas y tabla*.

| | | |
|--|--------------------------|--|
| Nombre del integrante: Alberto Pons Buttazzo | | |
| Descripción de la actividad: cambios de medidas en los pesos de kilogramos a gramos y eliminación de columnas de la tabla innecesarias. | | |
| Duración: 1 día | Esfuerzo: 1/2 día | Necesario para historias de usuario: 3 |

Tabla 41. Tarea 9.1: Cambios en vistas y tabla

| SPRINT BACKLOG | |
|---|--|
| Tareas | Metodologías |
| Tarea 9: Funcionalidades de las vistas para la web. | Metodología 9: Implementación de los 4 métodos para las gráficas con la variación de los pesos en el tiempo. |
| Tarea 9.1: Cambios en vistas y tabla. | Metodología 9.1: Cambio de las medidas de los pesos de kilogramos a gramos y eliminación de columnas de la tabla innecesarias. |
| SPRINT GOAL | |
| El objetivo del Sprint 3 es el de permitir al usuario visualizar de 4 formas distintas las gráficas con la variación de los pesos en el tiempo, y corregir medidas y columnas de las vistas y la tabla. | |

Tabla 42. Sprint 3 Backlog

| Estado | Fecha Inicio | Fecha Fin | Esfuerzo planificado | Duración planificada |
|-------------------|---------------|-----------|----------------------|----------------------|
| Trabajando en ... | dic. 19, 2020 | ene. 11 | 56 hours | 70hours |
| En espera | ene. 4 | ene. 4 | 4 hours | 5hours |
| Total | | | 60 hours | 75hours |

Ilustración 17. Sprint 3 - Sprint Backlog inicial

6.4.2. SPRINT 3 ANÁLISIS MEDIANTE CASOS DE USO

El siguiente diagrama de casos de uso de la ilustración 18, muestra las funcionalidades que se van a implementar en la web en el Sprint 3 (teniendo en cuenta la tarea 9.2, que aparece más adelante):

Aplicación web para la visualización de datos de sensor “Smart Bin”

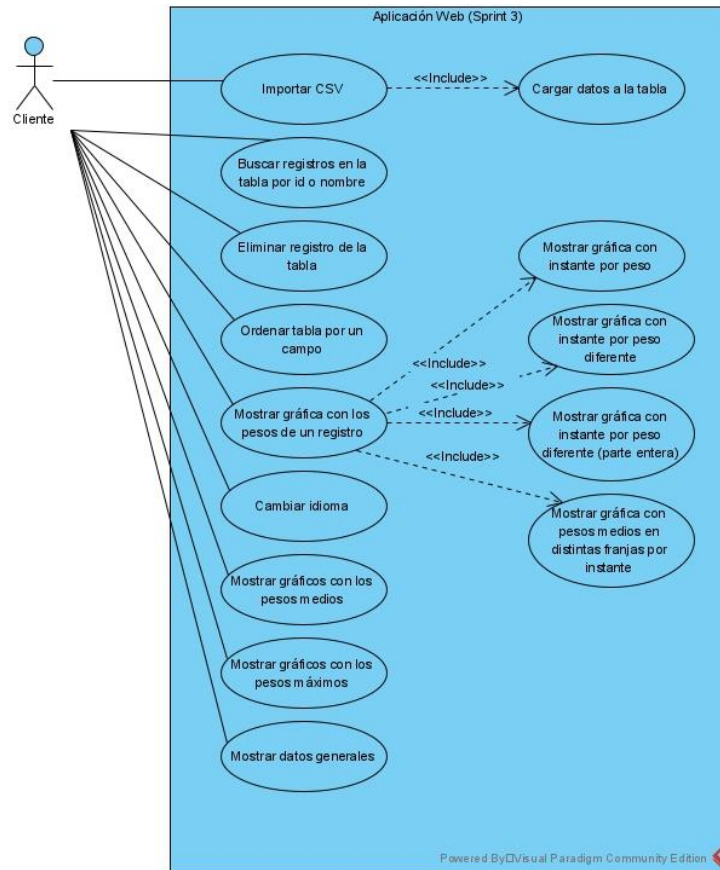


Ilustración 18. Diagrama casos de uso del Sprint 3

6.4.3. SPRINT 3 DISEÑO MEDIANTE UML

El siguiente diagrama UML de la ilustración 19, muestra la estructura Modelo-Vista-Controlador, junto a las funcionalidades implementadas en el Sprint 3 (se incluye la clase *ConfiguracionController* y varios métodos):

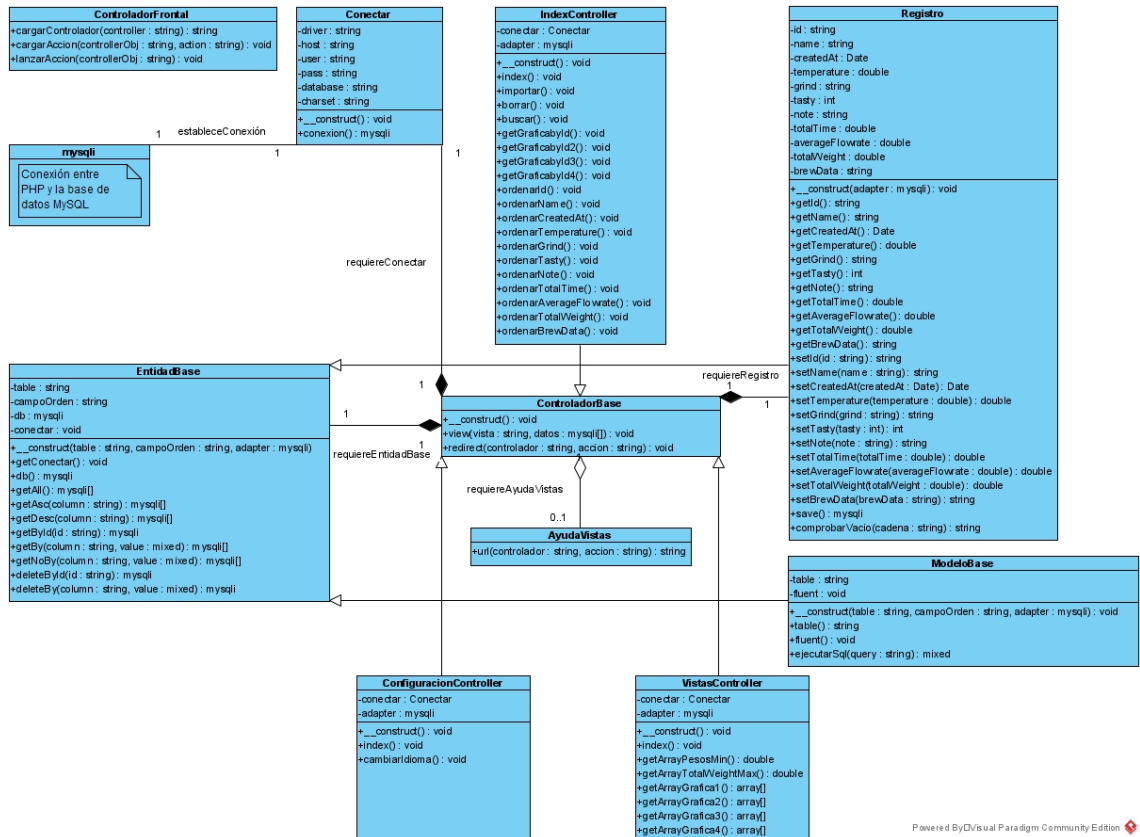


Ilustración 19. Diagrama UML del Sprint 3

6.4.4. SPRINT 3 DESARROLLO

En primer lugar, se establece una organización sobre la tarea 9 (*Funcionalidades de las vistas para la web*) en el primer Daily Scrum del Sprint:

- Permitir mediante un desplegable las 4 opciones para las gráficas.

Se crea un desplegable que permite seleccionar las siguientes opciones referentes a las gráficas con la variación de los pesos en el tiempo:

- Instante por peso.
- Instante por peso diferente.
- Instante por peso diferente en su parte entera.
- Peso medio en franjas de 10 instantes.
- Peso medio en franjas de 20 instantes.
- Peso medio en franjas de 50 instantes.

- Peso medio en franjas de 100 instantes.

Dependiendo de la opción seleccionada, se llama a uno de los 4 métodos del controlador *IndexController*. Además, es necesario el uso de un script para mantener la opción seleccionada al recargar la web (Karagiannis, 2014).

En segundo lugar, se organiza la tarea 9.1 (*Cambios en vistas y tabla*):

- Modificar medidas de kilogramos a gramos en todas las vistas.
- Eliminar columnas innecesarias.

Tras realizar los cambios en las medidas, se eliminan de la tabla las columnas: *temperature*, *grind*, *tasty* y *brewData*. La última columna *brewData* con todos los pesos es eliminada, ya que se refleja la variación de los mismos en las gráficas.

Finalmente, surge una nueva tarea en la tabla 44 (*Tarea 9.2: Internalización al inglés para la web*) a mitad del Sprint, que consiste en la traducción al inglés de toda la web. Permitted navegar en la web en inglés o en castellano. Por tanto, el Sprint Backlog y Sprint Goal se ven modificados en la tabla 45 e ilustración 20. Además una nueva historia de usuario surge como se puede observar en la tabla 43 (*Historia de usuario 3.1: Internalización al inglés*).

| Historia de usuario 3.1: Integración de ambas aplicaciones | |
|---|--|
| Rol | Cliente |
| Necesidad y beneficio | Permitir navegar en castellano o inglés |
| Criterio de aceptación (SMART) | La aplicación permite el cambio de idioma de castellano a inglés y viceversa |

Tabla 43. Internalización al inglés

| |
|--|
| Nombre del integrante: Alberto Pons Buttazzo |
| Descripción de la actividad: traducción de toda la web al inglés, permitiendo navegar en castellano o inglés. |

| | | |
|-------------------------|-------------------------|--|
| Duración: 4 días | Esfuerzo: 3 días | Necesario para historias de usuario: 3.1 |
|-------------------------|-------------------------|--|

Tabla 44. Tarea 9.2: Internalización al inglés

| SPRINT BACKLOG | |
|--|--|
| Tareas | Metodologías |
| Tarea 9: Funcionalidades de las vistas para la web. | Metodología 9: Implementación de los 4 métodos para las gráficas con la variación de los pesos en el tiempo. |
| Tarea 9.1: Cambios en vistas y tabla. | Metodología 9.1: Cambio de las medidas de los pesos de kilogramos a gramos y eliminación de columnas de la tabla innecesarias. |
| Tarea 9.2: Internalización al inglés. | Metodología 9.2: traducir todas las cadenas de texto en inglés y en base al idioma elegido, seleccionar unas cadenas u otras. |
| SPRINT GOAL | |
| El objetivo del Sprint 3 es el de permitir al usuario visualizar de 4 formas distintas las gráficas con la variación de los pesos en el tiempo, corregir medidas y columnas de las vistas y la tabla, y traducir la web al inglés. | |

Tabla 45. Sprint 3 Backlog intermedio

| Sprint 3 Intermedio: 1/2 Historia de usuario 3: Generación de vi... | Estado | Fecha Inicio | Fecha Fin | Esfuerzo planificado | Duración planificada |
|---|---------------------|---------------|-----------|----------------------|----------------------|
| Tarea 9: Funcionalidades de las vistas para la web | Listo para revis... | dic. 19, 2020 | ene. 11 | 56 hours | 70hours |
| Tarea 9.1: Cambios en vistas y tabla | Listo para revis... | ene. 4 | ene. 4 | 4 hours | 5hours |
| Tarea 9.2: Internalización al inglés | En espera | ene. 7 | ene. 11 | 20 hours | 25hours |
| + Agregar Activity | | | | | |
| | | | | 80 hours Total | 100hours Total |

Ilustración 20. Sprint 3 - Sprint Backlog intermedio

Por último, se planifica la tarea 9.2 (*Internalización al inglés*):

- Traducir todas las cadenas de texto al inglés.
- Generar botones y funcionalidad para cambiar de unas cadenas a otras.

Se comienza creando dos ficheros (*en* y *es*) con todas las cadenas de texto en castellano y en inglés. Seguidamente, se implementa una nueva vista *Configuración* y su controlador *ConfiguracionController* con un método que alterna entre un fichero con unas cadenas al otro. Finalmente, se generan dos botones que llaman a ese método, y en base a la elección traduce la web al castellano o al inglés (Matt, 2011).

6.4.5. SPRINT 3 REVIEW

Durante la reunión de revisión, en primer lugar se revisan los elementos del Sprint Backlog de la ilustración 21 que se han completado durante el desarrollo del tercer Sprint:

| Incremento de Sprint 3: 1/2 Historia de usuario 3: Generación d... | Estado | Fecha Inicio | Fecha Fin | Esfuerzo real | Duración real | |
|--|--------|---------------|-----------|-------------------|------------------|--|
| Tarea 9: Funcionalidades de las vistas para la web | Listo | dic. 19, 2020 | ene. 4 | 36 hours | 45hours | |
| Tarea 9.1: Cambios en vistas y tabla | Listo | ene. 4 | ene. 4 | 4 hours | 5hours | |
| Tarea 9.2: Internalización al inglés | Listo | ene. 7 | ene. 8 | 16 hours | 20hours | |
| + Agregar Activity | | | | | | |
| | | | | 56 hours Total | 70hours Total | |

Ilustración 21. Sprint 3 - Incremento

Seguidamente la reunión finaliza tras medir el progreso del proyecto mediante la gráfica de la ilustración 22:

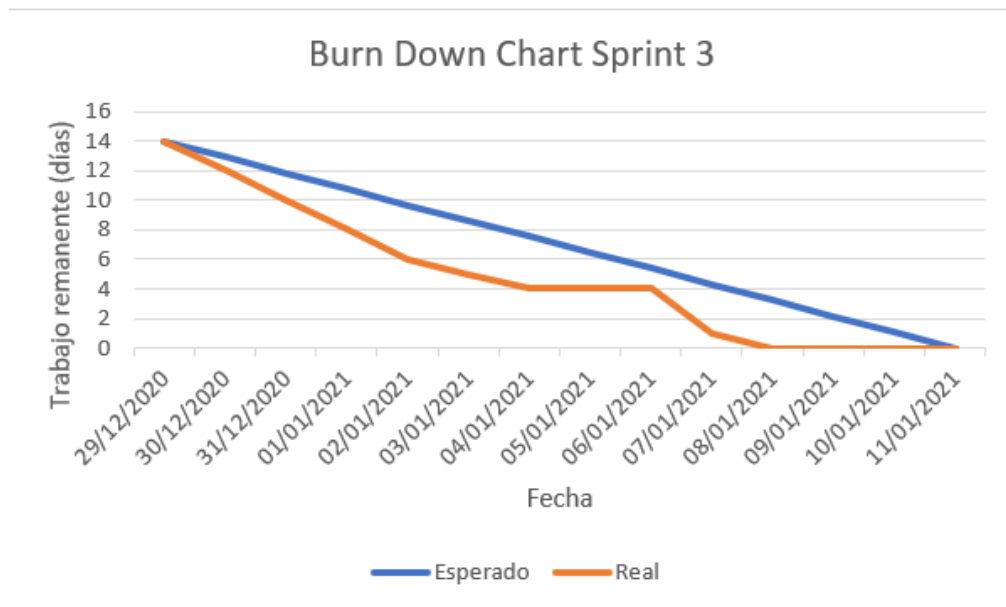


Ilustración 22. Burn Down Chart Sprint 3

6.4.6. SPRINT 3 RETROSPECTIVE

El enfoque inicial marcaba un desarrollo progresivo a lo largo de los días. En cambio, el enfoque final demuestra una progresión más adelantada que la esperada, incluso se acaba antes de lo planeado el día 8/01/2021 (en el enfoque inicial se estimó que finalizaría el 11/01/2021). Esto se debe al haber empezado a adelantar trabajo de la tarea 9 en el Sprint anterior. Las posibles mejoras en futuros Sprints sugieren otra vez un reparto de tareas más próximo a la realidad, ya que en este Sprint se podría haber adelantado trabajo del siguiente Sprint, si no fuera por la inclusión de dos nuevas tareas (9.1 y 9.2).

6.5. SPRINT 4 - EXTRAS FINALES + 2/2 HISTORIA DE USUARIO 3: GENERACIÓN DE VISTAS

6.5.1. SPRINT 4 PLANNING MEETING

Durante la reunión de planificación se establecen el Sprint Backlog y el Sprint Goal para el cuarto Sprint en la tabla 46 e ilustración 23.

| SPRINT BACKLOG | |
|---|--|
| Tareas | Metodologías |
| Tarea 10: Diseño sobre ventana de las vistas para la web. | Metodología 10: Implementar diseño sobre la ventana donde se muestran las vistas, haciendo uso de Bootstrap y de scripts que ofrezcan dinamismo a las gráficas. |
| Tarea 11: Funcionalidades extras restantes para la web. | Metodología 11: Crear funciones extras, tales como un apartado de configuración para las gráficas y la generación de una nueva gráfica que permita representar varias series en la variación de los pesos. |
| Tarea 12: Finalización de diseño para la web. | Metodología 12: Implementar cambios de diseño en la página de configuración, haciendo uso de más iconos, y cambios en la tabla, |

| | |
|---|--|
| | alterando orden de columnas y el alineamiento del texto. |
| SPRINT GOAL | |
| El objetivo del Sprint 4 es el de agregar una nueva gráfica multiserie, ofrecer opciones de configuración para las gráficas y realizar cambios de diseño que ofrezcan mayor usabilidad. | |

Tabla 46. Sprint 4 Backlog

| Sprint 4: Extras finales + 2/2 Historia de usuario 3: Generación ... | Estado | Fecha Inicio | Fecha Fin | Esfuerzo planificado | Duración planificada |
|--|-----------|--------------|-----------|----------------------------------|---------------------------------|
| Tarea 10: Diseño sobre ventana de las vistas para la web | En espera | ene. 12 | ene. 15 | 16 hours | 20hours |
| Tarea 11: Funcionalidades extras restantes para la web | En espera | ene. 16 | ene. 21 | 24 hours | 30hours |
| Tarea 12: Finalización de diseño para la web | En espera | ene. 22 | ene. 25 | 16 hours | 20hours |
| + Agregar Activity | | | | 56 hours <small>Total</small> | 70hours <small>Total</small> |

Ilustración 23. Sprint 4 - Sprint Backlog inicial

6.5.2. SPRINT 4 ANÁLISIS MEDIANTE CASOS DE USO

El siguiente diagrama de casos de uso de la ilustración 24, muestra las funcionalidades que se van a implementar en la web en el Sprint 4:

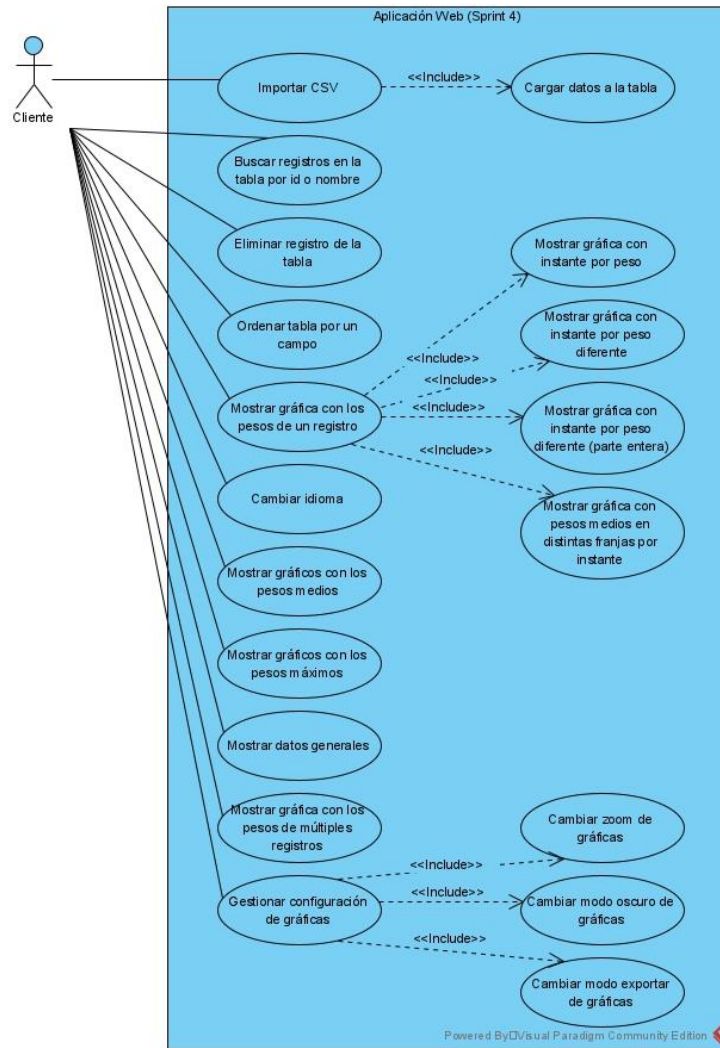


Ilustración 24. Diagrama casos de uso del Sprint 3

6.5.3. SPRINT 4 DISEÑO MEDIANTE UML

El siguiente diagrama UML de la ilustración 25, muestra la estructura Modelo-Vista-Controlador, junto a las funcionalidades implementadas en el Sprint 4 (no se incluyen clases, pero si varios métodos):

Aplicación web para la visualización de datos de sensor “Smart Bin”

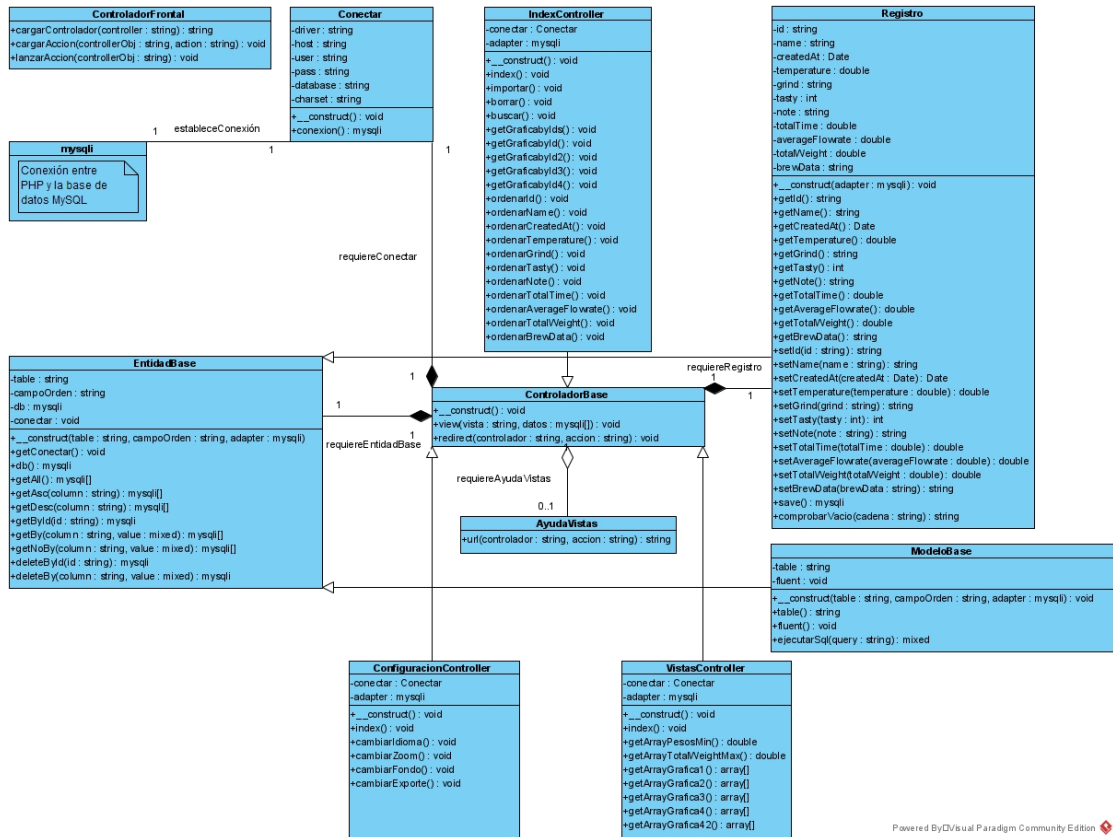


Ilustración 25. Diagrama UML del Sprint 4

6.5.4. SPRINT 4 DESARROLLO

En primer lugar, se establece una organización sobre la tarea 10 (*Diseño sobre ventana de las vistas para la web*) en el primer Daily Scrum del Sprint:

- Implementar diseño a las gráficas.

Se incluyen nuevas características de diseño sobre las gráficas, configurando los parámetros de zoom, exportación y del color de fondo de las mismas. Se podrán modificar mediante unas opciones de configuración que se exponen en la siguiente tarea.

En segundo lugar, se organiza la tarea 11 (*Funcionalidades extras restantes para la web*):

- Permitir habilitar o deshabilitar el zoom, exportación y modo oscuro en las gráficas.
- Generar una gráfica multiserie en la variación de los pesos en el tiempo, permitiendo seleccionar varios registros al usuario.

Se comienza implementando tres nuevos métodos en el controlador *ConfiguracionController*, que permiten alternar los parámetros de zoom, exportación y modo oscuro respectivamente.

Seguidamente, se crea un nuevo método *getGraficabylds* (en el controlador *IndexController*) que calcula los ejes X (instante de tiempo) e Y (peso en gramos) para la variación de los pesos por instante de tiempo para varios registros. Es la misma funcionalidad que tiene el método *getGraficabyld*, pero aplicada a varios registros, lo que genera a su vez varias series en la gráfica y permite contrastar las variaciones en los pesos de registros diferentes.

Finalmente, se planifica la tarea 12 (*Finalización de diseño para la web*):

- Incluir una nueva columna en la tabla para poder seleccionar varios registros.
- Alterar el orden de columnas para ofrecer una mayor usabilidad.
- Alinear en el centro el texto de todo la tabla.

Por último, se incluye una columna con casillas de verificación (*checkbox*) para poder seleccionar varios registros a la hora de generar la gráfica multiserie. Además, se alinea el texto de la tabla y se altera el orden de las columnas de la siguiente manera para ofrecer una vista más intuitiva:

- Selección de varios registros al inicio.
- Datos de los registros en el centro.
- Expansión a una gráfica en penúltimo lugar.
- Eliminación de un registro en último lugar.

6.5.5. SPRINT 4 REVIEW

Durante la reunión de revisión, en primer lugar se revisan los elementos del Sprint Backlog de la ilustración 26 que se han completado durante el desarrollo del cuarto Sprint:

Aplicación web para la visualización de datos de sensor “Smart Bin”

| Incremento de Sprint 4: Extras finales + 2/2 Historia de usuario ... | Estado | Fecha Inicio | Fecha Fin | Esfuerzo real | Duración real | |
|--|--------|--------------|-----------|----------------|---------------|--|
| Tarea 10: Diseño sobre ventana de las vistas para la web | Listo | ene. 12 | ene. 13 | 4 hours | 5hours | |
| Tarea 11: Funcionalidades extras restantes para la web | Listo | ene. 13 | ene. 16 | 16 hours | 20hours | |
| Tarea 12: Finalización de diseño para la web | Listo | ene. 18 | ene. 18 | 4 hours | 5hours | |
| + Agregar Activity | | | | | | |
| | | | | ene. 12 - 18 | ene. 13 - 18 | |
| | | | | 24 hours Total | 30hours Total | |

Ilustración 26. Sprint 4 - Incremento

Seguidamente la reunión finaliza tras medir el progreso del proyecto mediante la gráfica de la ilustración 27:

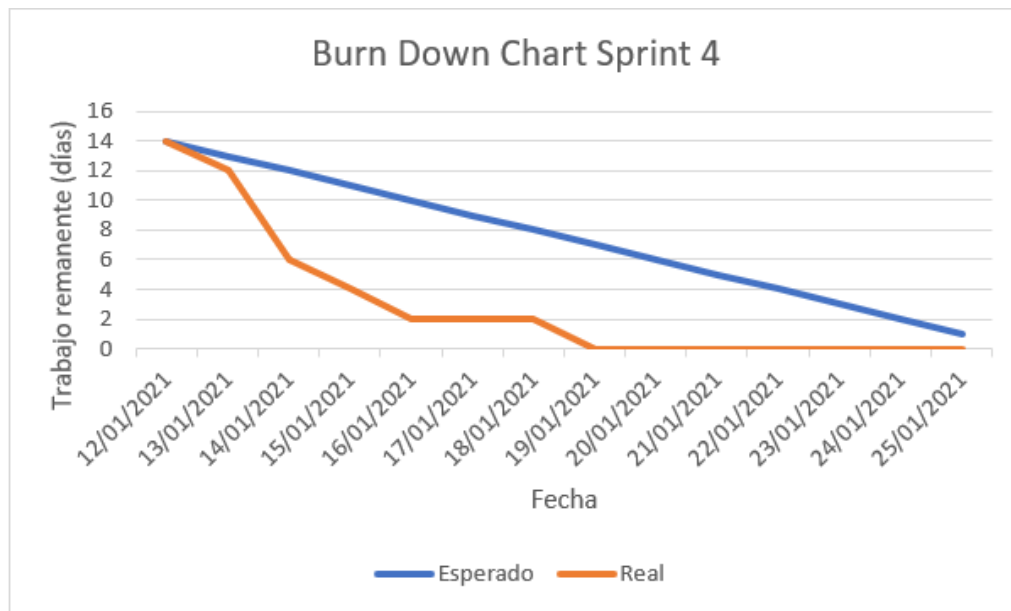


Ilustración 27. Burn Down Chart Sprint 4

6.5.6. SPRINT 4 RETROSPECTIVE

El enfoque inicial marcaba un desarrollo progresivo a lo largo de los días durante dos semanas. En cambio, el enfoque final supone una única semana de metaserie desarrollo y demuestra una progresión más adelantada que la esperada, incluso se acaba antes de lo planeado el día 18/01/2021 (en el enfoque inicial se estimó que finalizaría el 25/01/2021). Esto se debe al haber empezado a adelantar trabajo antes del comienzo de los exámenes finales. Con esto se puede concluir que una posible mejora para futuros Sprints consistiría en tener en mayor consideración factores externos, como en este caso han sido los exámenes finales.

6.6. SPRINT 5 - HISTORIA DE USUARIO 4: IMPLEMENTACIÓN DE ALGORITMOS

6.6.1. SPRINT 5 PLANNING MEETING

Durante la reunión de planificación se establecen el Sprint Backlog y el Sprint Goal para el quinto Sprint en la tabla 47 e ilustración 28.

| SPRINT BACKLOG | |
|---|---|
| Tareas | Metodologías |
| Tarea 13: Entorno de implementación para la app inteligente. | Metodología 13: Crear directorio de trabajo en Jupyter Notebook y Google Colab. |
| Tarea 14: Investigación inicial de Machine Learning con Arima y Prophet para la app inteligente. | Metodología 14: Empezar a usar las librerías Arima y Prophet sobre los datos del sensor de peso. |
| Tarea 15: Predicción de la variación de los pesos con Prophet | Metodología 15: Ejecutar el modelo Prophet sobre dos dataframes que reflejen los pesos de un experimento y sus instantes de toma. De esta manera, se generan métricas para evaluar la precisión en la predicción. |
| SPRINT GOAL | |
| El objetivo del Sprint 5 es el de generar unas predicciones iniciales con Prophet y Arima sobre la variación de los pesos en cada instante de tiempo, y en el caso de Prophet se empiezan a generar métricas para valorar la precisión del algoritmo. | |

Tabla 47. Sprint 5 Backlog

| Sprint 5: Historia de usuario 4: Implementación de algoritmos | Estado | Fecha Inicio | Fecha Fin | Esfuerzo planificado | Duración planificado |
|---|-----------|--------------|-----------|----------------------|----------------------|
| Tarea 13: Entorno de implementación para la app inteligente | En espera | feb. 2 | feb. 3 | 8 hours | 10hours |
| Tarea 14: Investigación inicial de Machine Learning con Arima y Prophet par... | En espera | feb. 4 | feb. 9 | 24 hours | 30hours |
| Tarea 15: Predicción de la variación de los pesos con Prophet para la app in... | En espera | feb. 10 | feb. 15 | 24 hours | 30hours |
| + Agregar Activity | | | | 56 hours Total | 70hours Total |

Ilustración 28. Sprint 5 - Sprint Backlog inicial

6.6.2. SPRINT 5 DESARROLLO

Durante el Sprint 5 no es necesario generar ningún diagrama de análisis o diseño por los siguientes dos motivos:

- Respecto al diagrama de casos de uso, no se va a implementar ninguna funcionalidad, sino que se van a iniciar unas primeras pruebas con Arima y Prophet sobre los pesos medidos por el sensor.
- Respecto al diagrama UML, hasta que no se integre la aplicación realizada en python sobre la aplicación web, no se incluyen nuevas clases o funciones.

Por lo tanto, se puede iniciar directamente con el desarrollo y establecer una organización sobre la tarea 13 (*Entorno de implementación para la app inteligente*):

- Crear repositorio en Jupyter Notebook.
- Crear repositorio en Google Colab.

Se va a trabajar con ambos entornos de manera online, sin necesidad de descargar ninguna herramienta o aplicación.

Posteriormente, se planifica la tarea 14 (*Investigación inicial de Machine Learning con Arima y Prophet para la app inteligente*):

- Importar librerías necesarias para usar Arima y Prophet.
- Leer CSV con los datos del sensor de peso.
- Crear dataframes iniciales para probar ambos modelos.
- Ejecutar los modelos por defecto.

Para poder comenzar a usar ambos modelos y poder leer del CSV, primero es necesario importar las librerías *Prophet*, *ARIMA* y *pandas* respectivamente. De esta manera, se recoge del CSV las columnas con la fecha y pesos medios.

Una vez se obtienen esos datos, se genera un dataframe (estructura de datos de dos dimensiones) para poder empezar a generar las predicciones con ambos modelos.

Con Prophet se establece un pronóstico de los siguientes 365 pesos medios modificando la variable *periods*. Por otro lado, se muestra una tabla con los pesos predichos (columna *yhat*) y se muestra una gráfica con el resultado usando la librería *plot*.

Se puede observar que los datos registrados en las ilustraciones 29 y 30 tienen una tendencia caótica. Esto se puede deber al reducido número de registros que se toman (13) y a la nula configuración del modelo.

| | ds | yhat |
|------------|---------------------|--------------|
| 373 | 2021-09-22 21:09:57 | 3580.019293 |
| 374 | 2021-09-23 21:09:57 | 11642.251659 |
| 375 | 2021-09-24 21:09:57 | 16978.694377 |
| 376 | 2021-09-25 21:09:57 | 2514.749102 |
| 377 | 2021-09-26 21:09:57 | 9141.728149 |

Ilustración 29. Tabla con pronóstico de Prophet de los pesos medios

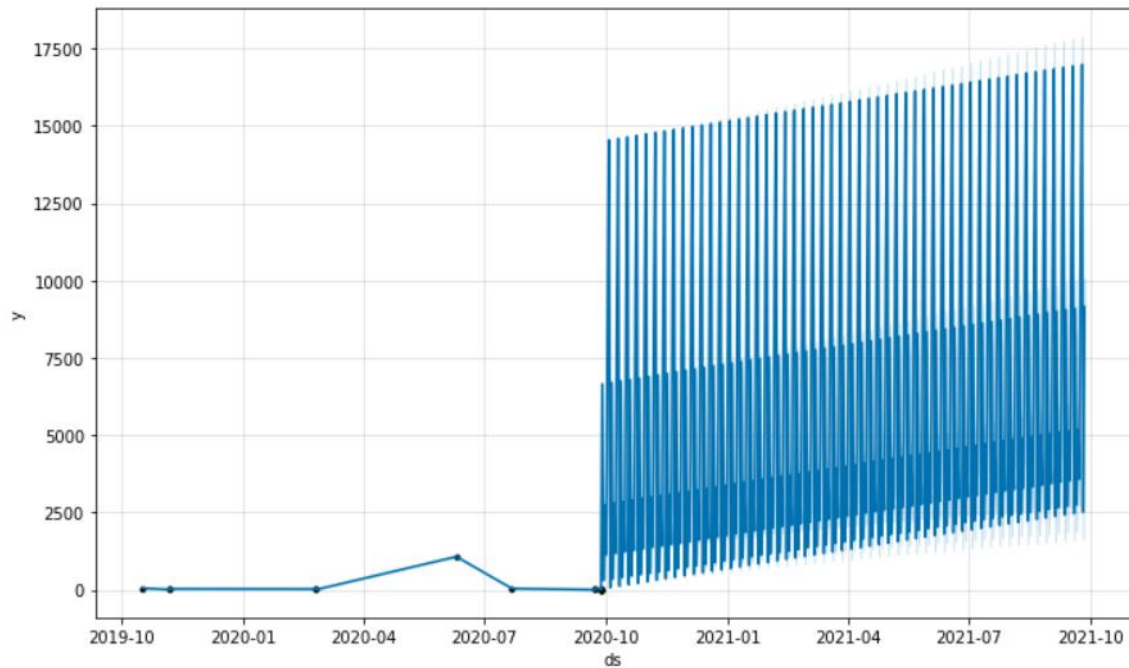


Ilustración 30. Gráfica con pronóstico de Prophet de los pesos medios

Con Arima se establece el conjunto de datos como los conjuntos *test* (juego de prueba) y *train* (donde se ajusta el modelo), y sus tamaños con la variable *size*. Por otro lado en la ilustración 31, se muestra una lista con los pesos predichos (variable *yhat*) y esperados (variable *obs*), se muestra una gráfica con el resultado usando la librería *pyplot* y se refleja el RSME con la variable *rmse* (error que marca la diferencia entre lo esperado y estimado).

```
predicted=1073.000459, expected=7.140000  
predicted=7.139535, expected=27.660000  
predicted=27.660233, expected=32.800000  
predicted=32.800018, expected=8.550000  
predicted=284.549725, expected=55.800000  
Test RMSE: 487.734
```

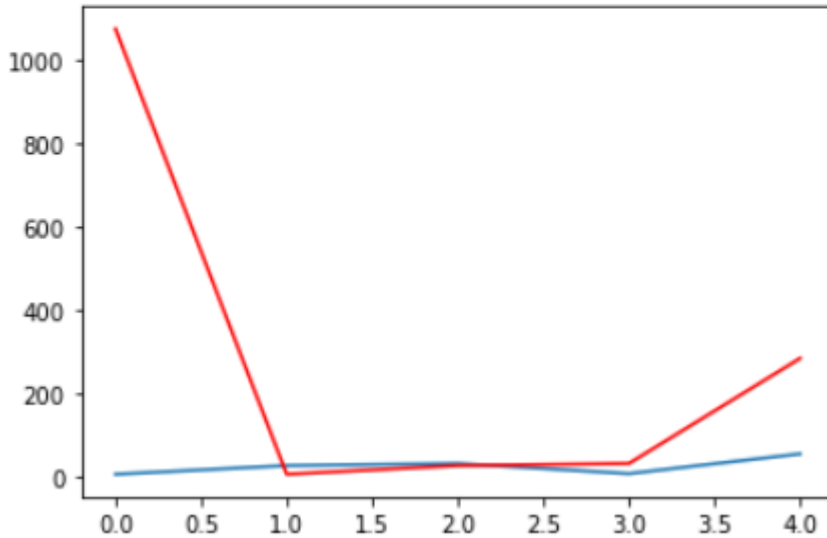


Ilustración 31. Pronóstico de Arima de los pesos medios

Adelantando lo que se realizará en el siguiente Sprint y en este mismo, se tratará de aumentar el número de registros de partida que se usan en el modelo. Esto será posible extrayendo todas las tomas de pesos de un registro en particular. Por otro lado, se empezará a configurar ambos modelos para optimizar las predicciones resultantes.

Por lo tanto, se puede iniciar directamente con el desarrollo y establecer una organización sobre la tarea 15 (*Predicción de la variación de los pesos con Prophet*):

- Generar dos dataframes de prueba sobre la variación de los pesos en cada instante.
- Usar el modelo de Prophet estándar (sin alterar sus parámetros de configuración) y otro alterando alguno de sus parámetros.
- Generar métricas para evaluar la precisión del algoritmo.

Se generan dos dataframes, ya que tras una investigación previa se prevé que suavizar la curva sobre la variación de los pesos puede ofrecer unos

resultados predictivos más precisos. De esta manera, se generan los siguientes dataframes:

- Uno donde se asocia un peso por instante, como se puede observar en la ilustración 32.
- Otro donde se asocia la media de pesos en un segundo a cada segundo, como se puede observar en la ilustración 33.

| | ds | y |
|------|----------------------------|------|
| 0 | 2020-02-24 18:42:50.197344 | 0.00 |
| 1 | 2020-02-24 18:42:50.394688 | 0.00 |
| 2 | 2020-02-24 18:42:50.592033 | 0.00 |
| 3 | 2020-02-24 18:42:50.789377 | 0.00 |
| 4 | 2020-02-24 18:42:50.986721 | 0.00 |
| ... | ... | ... |
| 4889 | 2020-02-24 18:58:55.013279 | 4.30 |
| 4890 | 2020-02-24 18:58:55.210623 | 4.30 |
| 4891 | 2020-02-24 18:58:55.407967 | 4.30 |
| 4892 | 2020-02-24 18:58:55.605312 | 4.30 |
| 4893 | 2020-02-24 18:58:55.802656 | 4.30 |

Ilustración 32. Dataframe con los pesos por cada instante

| | ds | y |
|-----|---------------------|-----|
| 0 | 2020-02-24 18:42:51 | 0.0 |
| 1 | 2020-02-24 18:42:52 | 0.0 |
| 2 | 2020-02-24 18:42:53 | 0.0 |
| 3 | 2020-02-24 18:42:54 | 0.0 |
| 4 | 2020-02-24 18:42:55 | 0.0 |
| ... | ... | ... |
| 961 | 2020-02-24 18:58:52 | 4.3 |
| 962 | 2020-02-24 18:58:53 | 4.3 |
| 963 | 2020-02-24 18:58:54 | 4.3 |
| 964 | 2020-02-24 18:58:55 | 4.3 |
| 965 | 2020-02-24 18:58:56 | 4.3 |

Ilustración 33. Dataframe con los pesos medios por cada segundo

Una vez definidos ambos dataframes, estos se crean y usan sobre los dos modelos de Prophet:

- Un modelo Prophet sin alterar.
- Otro modelo modificando los siguientes valores (Prophet, 2021):

- *n_changepoints* (por defecto a 25): valor que se incrementa hasta 45 para aumentar los puntos de cambio en la tendencia que toma el algoritmo.
- *changepoint_prior_scale* (por defecto a 0.05): valor que se incrementa hasta 0.1 para aumentar la flexibilidad de la tendencia.
- *changepoint_range* (por defecto a 0.8): valor que se incrementa hasta 0.9 para esparcir los puntos de cambio (*changepoints*) sobre el primer 90% de la serie.

Por último, se generan una serie de métricas para evaluar la precisión del algoritmo (Brown E. , 2018) (ZeroSpectrum, 2019):

- R-Squared: establece el porcentaje de variación que el modelo es capaz de explicar.
- MSE (Mean Squared Error): al realizar el cuadrado se obtiene un mayor peso en los cambios más distantes entre los valores actuales y predichos, obteniendo el error medio al cuadrado.
- MAE (Mean Absolute Error): establece la media del cambio entre los valores actuales y predichos, obteniendo el error medio absoluto.
- RMSE (Root Mean Squared Error): al realizar la raíz cuadrada sobre el MSE, se obtiene el error medio.

En estas primeras pruebas, se obtienen los siguientes resultados en la tabla 48:

| | R-Squared | MAE | MSE | RMSE |
|--|-----------|-------|-------|-------|
| Primer dataframe con el modelo por defecto | 0.611 | 0.158 | 0.581 | 0.762 |

| | | | | |
|---|-------|-------|-------|-------|
| Segundo dataframe con el modelo por defecto | 0.629 | 0.132 | 0.555 | 0.745 |
| Primer dataframe con el modelo alterado | 0.798 | 0.168 | 0.226 | 0.476 |
| Segundo dataframe con el modelo alterado | 0.823 | 0.138 | 0.199 | 0.446 |

Tabla 48. Primeros resultados de Prophet

Los resultados ofrecen una mayor precisión predictiva con el segundo dataframe con el modelo alterado, ya que se reducen todos los errores y se aumenta la varianza que el modelo es capaz de explicar.

Por otro lado, se puede observar una tendencia ligeramente ascendente en la predicción de la cuarta opción en la gráfica de la ilustración 34 (*Segundo dataframe con el modelo alterado*):

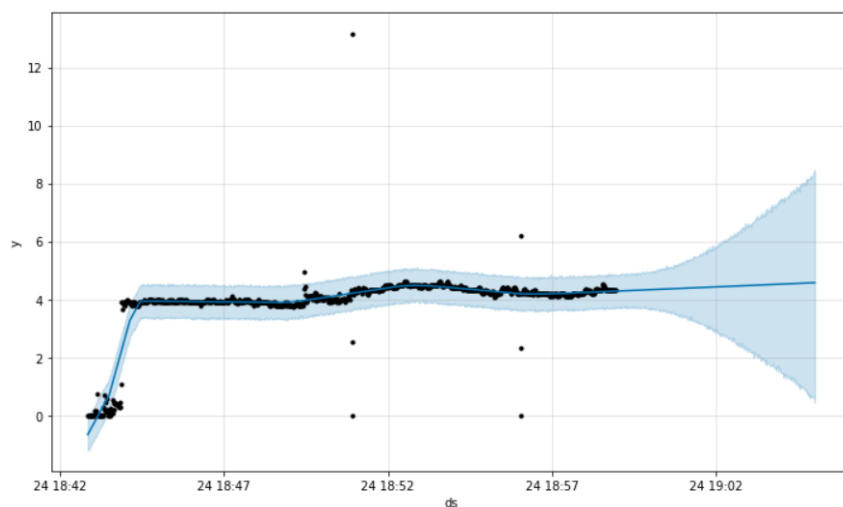


Ilustración 34. Predicción con Prophet

6.6.3. SPRINT 5 REVIEW

Durante la reunión de revisión, en primer lugar se revisan los elementos del Sprint Backlog de la ilustración 35 que se han completado durante el desarrollo del quinto Sprint:

| Incremento de Sprint 5: Historia de usuario 4: Implementación ... | Estado | Fecha Inicio | Fecha Fin | Esfuerzo real | Duración real |
|---|--------|--------------|-----------|-------------------|------------------|
| Tarea 13: Entorno de implementación para la app inteligente | Listo | feb. 2 | feb. 3 | 8 hours | 10hours |
| Tarea 14: Investigación inicial de Machine Learning con Arima y Prophet par... | Listo | feb. 4 | feb. 8 | 20 hours | 25hours |
| Tarea 15: Predicción de la variación de los pesos con Prophet para la app in... | Listo | feb. 9 | feb. 13 | 20 hours | 25hours |
| + Agregar Activity | | | | | |
| | | | | 48 hours Total | 60hours Total |

Ilustración 35. Sprint 5 - Incremento

Seguidamente la reunión finaliza tras medir el progreso del proyecto mediante la gráfica de la ilustración 36:

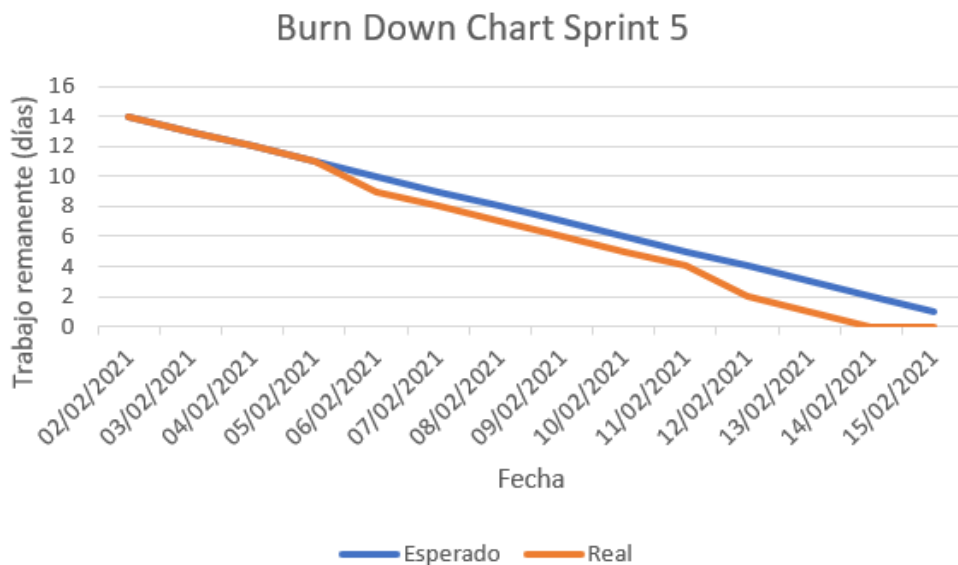


Ilustración 36. Burn Down Chart Sprint 5

6.6.4. SPRINT 5 RETROSPECTIVE

El enfoque inicial marcaba un desarrollo progresivo a lo largo de los días durante dos semanas. En cambio, el enfoque final demuestra una progresión más adelantada que la esperada, incluso se acaba antes de lo planeado el día 13/02/2021 (en el enfoque inicial se estimó que finalizaría el 15/02/2021). Aunque la planificación es muy similar a lo que realmente se ha trabajado, con lo que no se estima la puntualización de ninguna mejora.

6.7. SPRINT 6 - HISTORIA DE USUARIO 5: GENERACIÓN DE PREDICCIONES

6.7.1. SPRINT 6 PLANNING MEETING

Durante la reunión de planificación se establecen el Sprint Backlog y el Sprint Goal para el sexto Sprint en la tabla 49 e ilustración 37.

| SPRINT BACKLOG | |
|--|---|
| Tareas | Metodologías |
| Tarea 16: Predicción de la variación de los pesos con Arima | Metodología 16: Ejecutar el modelo Arima sobre dos dataframes que reflejen los pesos de un experimento y sus instantes de toma. De esta manera, se generan métricas para evaluar la precisión en la predicción. |
| Tarea 17: Predicción final de la variación de los pesos con Prophet para la app inteligente | Metodología 17: Ejecutar el modelo Prophet sobre todos los experimentos y contrastando las métricas generadas mediante múltiples variaciones sobre los parámetros del algoritmo |
| SPRINT GOAL | |
| El objetivo del Sprint 6 es el de generar unas métricas con Arima y terminar de optimizar la precisión en las predicciones de Prophet. | |

Tabla 49. Sprint 6 Backlog

| Estado | Fecha Inicio | Fecha Fin | Esfuerzo esperado | Duración esperada |
|-----------|--------------|-----------|-------------------|-------------------|
| En espera | feb. 16 | feb. 22 | 28 hours | 35hours |
| En espera | feb. 23 | mar. 1 | 28 hours | 35hours |
| | | | 56 hours Total | 70hours Total |

Ilustración 37. Sprint 6 - Sprint Backlog inicial

6.7.2. SPRINT 6 DESARROLLO

Durante el Sprint 6 tampoco es necesario generar ningún diagrama de análisis o diseño por los siguientes dos motivos:

- Respecto al diagrama de casos de uso, no se va a implementar ninguna funcionalidad, sino que se van a seguir haciendo pruebas con Arima y Prophet sobre los pesos medidos por el sensor.
- Respecto al diagrama UML, hasta que no se integre la aplicación realizada en python sobre la aplicación web, no se incluyen nuevas clases o funciones.

En primer lugar, se organiza la tarea 16 (*Predicción de la variación de los pesos con Arima*):

- Reutilizar los dos dataframes de prueba que se usaron con Prophet.
- Usar el modelo de Arima autogenerado.
- Generar métricas para evaluar la precisión del algoritmo.

Se comienza transformando los dataframes en series, es decir, que Arima debe usar una estructura de datos unidimensional, en vez de una bidimensional.

Seguidamente, se decide usar la opción de Auto Arima (Singh, 2018), que permite generar los parámetros del modelo de forma automática. Esta alternativa permite simplificar la evaluación de los datos y reducir significativamente el tiempo de ejecución del modelo. Los parámetros que se autogeneran son los siguientes:

- p : orden del proceso autorregresivo.
- d : orden del proceso diferencial.
- q : orden de la media móvil (promedios de un subconjunto de datos).

Finalmente, en estas primeras pruebas se obtienen los siguientes resultados de la tabla 50:

| | MAE | MSE | RMSE |
|-------------------------|-------|-------|-------|
| Primer dataframe con el | 1.226 | 2.115 | 1.454 |

| | | | |
|--|-------|-------|-------|
| modelo autogenerado | | | |
| Segundo dataframe con el modelo autogenerado | 0.181 | 0.123 | 0.350 |

Tabla 50. Primeros resultados de Arima

Los resultados ofrecen una mayor precisión predictiva con el segundo dataframe al igual que con Prophet, ya que de la misma manera se reducen todos los errores.

Por otro lado, se puede observar una tendencia ligeramente ascendente en la predicción de la segunda opción en la gráfica de la ilustración 38 (*Segundo dataframe con el modelo autogenerado*):

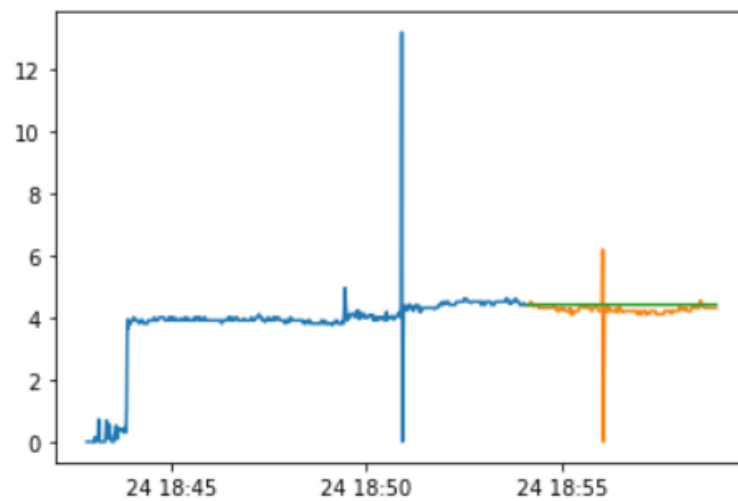


Ilustración 38. Predicción con Arima

Por último, se planifica la tarea 17 (Predicción final de la variación de los pesos con Prophet para la app inteligente):

- Se ejecuta el modelo Prophet con todos los experimentos.
- Se parte del dataframe alterado (con los pesos medios por cada segundo), pero realizando múltiples variaciones sobre los parámetros del algoritmo (*n_changepoints*, *changepoint_prior_scale* y *changepoint_range*).

- Generar predicciones y métricas para cada uno de los casos.

Hasta el momento se han realizado las pruebas con el registro con identificador *GihlPyhIFh*, pero en esta ocasión se va a experimentar con los 8 registros disponibles.

A continuación, se va a mostrar varias tablas (de la tabla 51 a la 58) con cada registro y las modificaciones que se han efectuado sobre los parámetros del algoritmo de predicción.

| Registro con identificador eHMc1is7KT con 34 pesos | | | | |
|---|------------------|------------|------------|-------------|
| Configuración del algoritmo | R-Squared | MSE | MAE | RMSE |
| Changepoint_prior_scale=0,05, changepoint_range=0,80 y n_changepoints=25 (Configuración por defecto) | 0,309 | 35449,852 | 155,901 | 188,281 |
| Changepoint_prior_scale=0,15 | 0,839 | 8252,498 | 70,074 | 90,843 |
| Changepoint_prior_scale=0,3 | 0,961 | 1968,392 | 35,087 | 44,366 |
| Changepoint_range=0,85 | 0,309 | 35449,838 | 155,890 | 188,281 |
| Changepoint_range=0,90 | 0,309 | 35449,977 | 155,773 | 188,281 |
| N_changepoints= 1/20 del nº de pesos | 0,309 | 35450,240 | 155,985 | 188,282 |
| N_changepoints= 1/10 del nº de pesos | 0,309 | 35449,853 | 155,918 | 188,281 |

Tabla 51. Resultados finales de Prophet: registro eHMc1is7KT

| Registro con identificador JmJ2por3Xk con 316 pesos | | | | |
|---|------------------|------------|------------|-------------|
| Configuración del algoritmo | R-Squared | MSE | MAE | RMSE |
| Changepoint_prior_scale=0,05, changepoint_range=0,80 y n_changepoints=25 (Configuración por defecto) | 0,626 | 643,209 | 18,854 | 25,361 |
| Changepoint_prior_scale=0,15 | 0,718 | 483,581 | 11,420 | 21,990 |
| Changepoint_prior_scale=0,3 | 0,725 | 471,400 | 10,532 | 21,711 |
| Changepoint_range=0,85 | 0,636 | 624,745 | 18,158 | 24,994 |
| Changepoint_range=0,90 | 0,616 | 659,535 | 19,250 | 25,681 |
| N_changepoints= 1/20 del nº de pesos | 0,626 | 642,911 | 18,733 | 25,355 |
| N_changepoints= 1/10 del nº de pesos | 0,629 | 637,822 | 18,644 | 25,255 |

Tabla 52. Resultados finales de Prophet: registro JmJ2por3Xk

| Registro con identificador fAFmRC5wb7 con 674 pesos | | | | |
|---|------------------|------------|------------|-------------|
| Configuración del algoritmo | R-Squared | MSE | MAE | RMSE |
| Changepoint_prior_scale=0,05, changepoint_range=0,80 y n_changepoints=25 (Configuración por defecto) | 0,566 | 24,667 | 1,662 | 4,966 |
| Changepoint_prior_scale=0,15 | 0,891 | 6,157 | 0,685 | 2,481 |
| Changepoint_prior_scale=0,3 | 0,900 | 5,649 | 0,567 | 2,371 |
| Changepoint_range=0,85 | 0,555 | 25,266 | 1,698 | 5,026 |
| Changepoint_range=0,90 | 0,492 | 28,831 | 1,873 | 5,369 |
| N_changepoints= 1/20 del nº de pesos | 0,528 | 26,777 | 1,836 | 5,174 |
| N_changepoints= 1/10 del nº de pesos | 0,537 | 26,307 | 1,707 | 5,129 |

Tabla 53. Resultados finales de Prophet: registro fAFmRC5wb7

| Registro con identificador AHwGDvXm1Y con 716 pesos | | | | |
|---|------------------|------------|------------|-------------|
| Configuración del algoritmo | R-Squared | MSE | MAE | RMSE |
| Changepoint_prior_scale=0,05, changepoint_range=0,80 y n_changepoints=25 (Configuración por defecto) | 0,960 | 3,416 | 0,815 | 1,848 |
| Changepoint_prior_scale=0,15 | 0,992 | 0,603 | 0,235 | 0,776 |
| Changepoint_prior_scale=0,3 | 0,993 | 0,538 | 0,230 | 0,734 |
| Changepoint_range=0,85 | 0,960 | 3,425 | 0,812 | 1,850 |
| Changepoint_range=0,90 | 0,958 | 3,573 | 0,857 | 1,890 |
| N_changepoints= 1/20 del nº de pesos | 0,964 | 3,084 | 0,758 | 1,756 |
| N_changepoints= 1/10 del nº de pesos | 0,963 | 3,137 | 0,767 | 1,771 |

Tabla 54. Resultados finales de Prophet: registro AHwGDvXm1Y

| Registro con identificador S3cQCqyeJc con 942 pesos | | | | |
|---|------------------|------------|------------|-------------|
| Configuración del algoritmo | R-Squared | MSE | MAE | RMSE |
| Changepoint_prior_scale=0,05, changepoint_range=0,80 y n_changepoints=25 (Configuración por defecto) | 0,922 | 2,946 | 0,948 | 1,716 |
| Changepoint_prior_scale=0,15 | 0,968 | 1,195 | 0,433 | 1,093 |
| Changepoint_prior_scale=0,3 | 0,973 | 0,998 | 0,358 | 0,999 |
| Changepoint_range=0,85 | 0,920 | 3,032 | 0,967 | 1,741 |
| Changepoint_range=0,90 | 0,921 | 2,997 | 0,966 | 1,731 |
| N_changepoints= 1/20 del nº de pesos | 0,926 | 2,788 | 0,916 | 1,669 |
| N_changepoints= 1/10 del nº de pesos | 0,932 | 2,561 | 0,865 | 1,600 |

Tabla 55. Resultados finales de Prophet: registro S3cQCqyeJc

| Registro con identificador GHIPIyHfH con 966 pesos | | | | |
|---|------------------|------------|------------|-------------|
| Configuración del algoritmo | R-Squared | MSE | MAE | RMSE |
| Changepoint_prior_scale=0,05, changepoint_range=0,80 y n_changepoints=25 (Configuración por defecto) | 0,795 | 0,225 | 0,167 | 0,475 |
| Changepoint_prior_scale=0,15 | 0,815 | 0,204 | 0,141 | 0,452 |
| Changepoint_prior_scale=0,3 | 0,822 | 0,196 | 0,135 | 0,443 |
| Changepoint_range=0,85 | 0,796 | 0,225 | 0,162 | 0,474 |
| Changepoint_range=0,90 | 0,796 | 0,225 | 0,163 | 0,475 |
| N_changepoints= 1/20 del nº de pesos | 0,800 | 0,221 | 0,161 | 0,470 |
| N_changepoints= 1/10 del nº de pesos | 0,802 | 0,219 | 0,160 | 0,468 |

Tabla 56. Resultados finales de Prophet: registro GHIPIyHfH

| Registro con identificador kSqTilqCT4 con 1398 pesos | | | | |
|---|------------------|------------|------------|-------------|
| Configuración del algoritmo | R-Squared | MSE | MAE | RMSE |
| Changepoint_prior_scale=0,05, changepoint_range=0,80 y n_changepoints=25 (Configuración por defecto) | 0,901 | 87,737 | 4,536 | 9,366 |
| Changepoint_prior_scale=0,15 | 0,911 | 78,814 | 3,985 | 8,877 |
| Changepoint_prior_scale=0,3 | 0,919 | 71,475 | 3,384 | 8,454 |
| Changepoint_range=0,85 | 0,901 | 87,716 | 4,597 | 9,365 |
| Changepoint_range=0,90 | 0,902 | 86,219 | 4,571 | 9,285 |
| N_changepoints= 1/20 del nº de pesos | 0,902 | 86,652 | 4,466 | 9,308 |
| N_changepoints= 1/10 del nº de pesos | 0,902 | 86,47 | 4,466 | 9,298 |

Tabla 57. Resultados finales de Prophet: registro kSqTilqCT4

| Registro con identificador UDaUHwzMye y con 1598 pesos | | | | |
|---|------------------|------------|------------|-------------|
| Configuración del algoritmo | R-Squared | MSE | MAE | RMSE |
| Changepoint_prior_scale=0,05, changepoint_range=0,80 y n_changepoints=25 (Configuración por defecto) | 0,793 | 0,346 | 0,158 | 0,588 |
| Changepoint_prior_scale=0,15 | 0,817 | 0,305 | 0,142 | 0,553 |
| Changepoint_prior_scale=0,3 | 0,822 | 0,298 | 0,141 | 0,546 |
| Changepoint_range=0,85 | 0,794 | 0,345 | 0,161 | 0,587 |
| Changepoint_range=0,90 | 0,795 | 0,342 | 0,160 | 0,585 |
| N_changepoints= 1/20 del nº de pesos | 0,796 | 0,341 | 0,154 | 0,584 |
| N_changepoints= 1/10 del nº de pesos | 0,796 | 0,341 | 0,153 | 0,584 |

Tabla 58. Resultados finales de Prophet: registro UDaUHwzMye

De estas pruebas se deduce que el único parámetro que ofrece una mejoría notable sobre las métricas es *changepoint_prior_scale*, que como se comentó con anterioridad, influye en la flexibilidad de la tendencia.

Además, dado que la diferencia en el valor de las métricas no varía demasiado entre un *changepoint_prior_scale* de 0,15 y 0,3, se entiende que a partir de cierto punto, aunque se aumente el valor, la mejoría no va a ser destacable.

De las dos conclusiones anteriores, se decide que la configuración definitiva del algoritmo va a ser manteniendo los valores base de los parámetros *changepoint_range* (0,8) y *n_changepoints* (25), y modificando el parámetro *changepoint_prior_scale* a 0,3.

Con eso se obtienen errores más pequeños y un porcentaje de variación que el modelo es capaz de explicar (R-Squared) que oscila entre el 72,5% (0,725) y el 99,3% (0,993), y que en la mayoría de casos es superior al 80% (0,8).

6.7.3. SPRINT 6 REVIEW

Durante la reunión de revisión, en primer lugar se revisan los elementos del Sprint Backlog de la ilustración 39 que se han completado durante el desarrollo del sexto Sprint:

| Incremento de Sprint 6: Historia de usuario 5: Generación de pr... | Estado | Fecha Inicio | Fecha Fin | Esfuerzo real | Duración real |
|--|--------|--------------|--------------|-------------------|------------------|
| Tarea 16: Predicción de la variación de los pesos con Arima para la app inte... | Listo | feb. 16 | feb. 19 | 16 hours | 20hours |
| Tarea 17: Predicción final de la variación de los pesos con Prophet para la a... | Listo | feb. 22 | feb. 26 | 20 hours | 25hours |
| + Agregar Activity | | | | | |
| | | feb. 16 - 22 | feb. 19 - 26 | 36 hours Total | 45hours Total |

Ilustración 39. Sprint 6 - Incremento

Seguidamente la reunión finaliza tras medir el progreso del proyecto mediante la gráfica de la Ilustración 30:

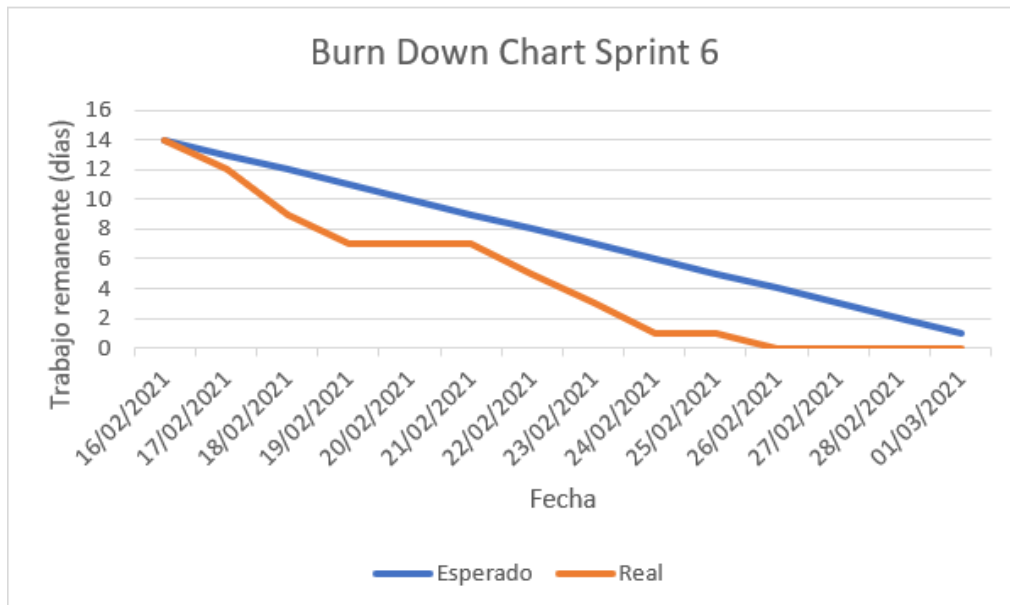


Ilustración 40. Burn Down Chart Sprint 6

6.7.4. SPRINT 6 RETROSPECTIVE

El enfoque inicial marcaba un desarrollo progresivo a lo largo de los días durante dos semanas. En cambio, el enfoque final demuestra una progresión más adelantada que la esperada, incluso se acaba antes de lo planeado el día 26/02/2021 (en el enfoque inicial se estimó que finalizaría el 01/03/2021). En este Sprint no se tuvo en cuenta la estructura de pruebas que se generó en el anterior Sprint, lo cual sirvió de punto de partida para las pruebas de este Sprint. Y por tanto, redujo el tiempo para elaborar estas pruebas. Un aspecto a mejorar para

futuros Sprints sería estimar cuanto tiempo se puede reducir la carga de trabajo con lo que se realizó en anteriores Sprints.

6.8. SPRINT 7 - HISTORIA DE USUARIO 6: INTEGRACIÓN DE AMBAS APLICACIONES

6.8.1. SPRINT 7 PLANNING MEETING

Durante la reunión de planificación se establecen el Sprint Backlog y el Sprint Goal para el séptimo Sprint en la tabla 59 e ilustración 41.

| SPRINT BACKLOG | |
|--|--|
| Tareas | Metodologías |
| Tarea 18: Entorno de implementación para la integración | Metodología 18: Descargar Anaconda Navigator para establecer un entorno con las librerías necesarias para ejecutar la aplicación inteligente de forma local. |
| Tarea 19: Conexión de ambas aplicaciones para la integración | Metodología 19: Conectar la aplicación inteligente con la aplicación web mediante el intérprete de comandos. |
| Tarea 20: Datos de entrada para la integración | Metodología 20: Generar los arrays con los pesos que se deben enviar a la aplicación inteligente. |
| Tarea 21: Modificación de la app inteligente para la integración | Metodología 21: Modificar aplicación inteligente para poder recibir los arrays con los pesos de la aplicación web y poder enviar los resultados predichos. |
| Tarea 22: Datos de salida para la integración | Metodología 22: Generar los arrays con los pesos y las métricas que se deben enviar a la aplicación web. |
| Tarea 23: Generación de vistas para la integración | Metodología 23: Generar gráfica de salida para poder visualizar los datos |

Aplicación web para la visualización de datos de sensor “Smart Bin”

| | |
|---|---|
| | recibidos por la aplicación inteligente (pesos y métricas). |
| SPRINT GOAL | |
| El objetivo del Sprint 7 es el de integrar la aplicación inteligente con la aplicación web, para así poder visualizar las predicciones a través de los gráficos de la web.. | |

Tabla 59. Sprint 7 Backlog

| ✔ Sprint 7: Historia de usuario 6: Integración de ambas aplicacion... | Estado | Fecha Inicio | Fecha Fin | Esfuerzo esperado | Duración esperada | + |
|---|-----------|--------------|-----------|-------------------|-------------------|---|
| Tarea 18: Entorno de implementación para la integración | En espera | mar. 2 | mar. 3 | 8 hours | 10hours | |
| Tarea 19: Conexión de ambas aplicaciones para la integración | En espera | mar. 4 | mar. 8 | 20 hours | 25hours | |
| Tarea 20: Datos de entrada para la integración | En espera | mar. 9 | mar. 10 | 8 hours | 10hours | |
| Tarea 21: Modificación de la app inteligente para la integración | En espera | mar. 11 | mar. 12 | 8 hours | 10hours | |
| Tarea 22: Datos de salida para la integración | En espera | mar. 13 | mar. 13 | 4 hours | 5hours | |
| Tarea 23: Generación de vistas para la integración | En espera | mar. 14 | mar. 15 | 8 hours | 10hours | |
| + Agregar Activity | | | | | | |
| | | mar. 2 - 14 | | mar. 3 - 15 | | |
| | | | | 56 hours Total | 70hours Total | |

Ilustración 41. Sprint 7 - Sprint Backlog inicial

6.8.2. SPRINT 7 ANÁLISIS MEDIANTE CASOS DE USO

El siguiente diagrama de casos de uso de la Ilustración 42, muestra las funcionalidades que se van a implementar en la web en el Sprint 7:

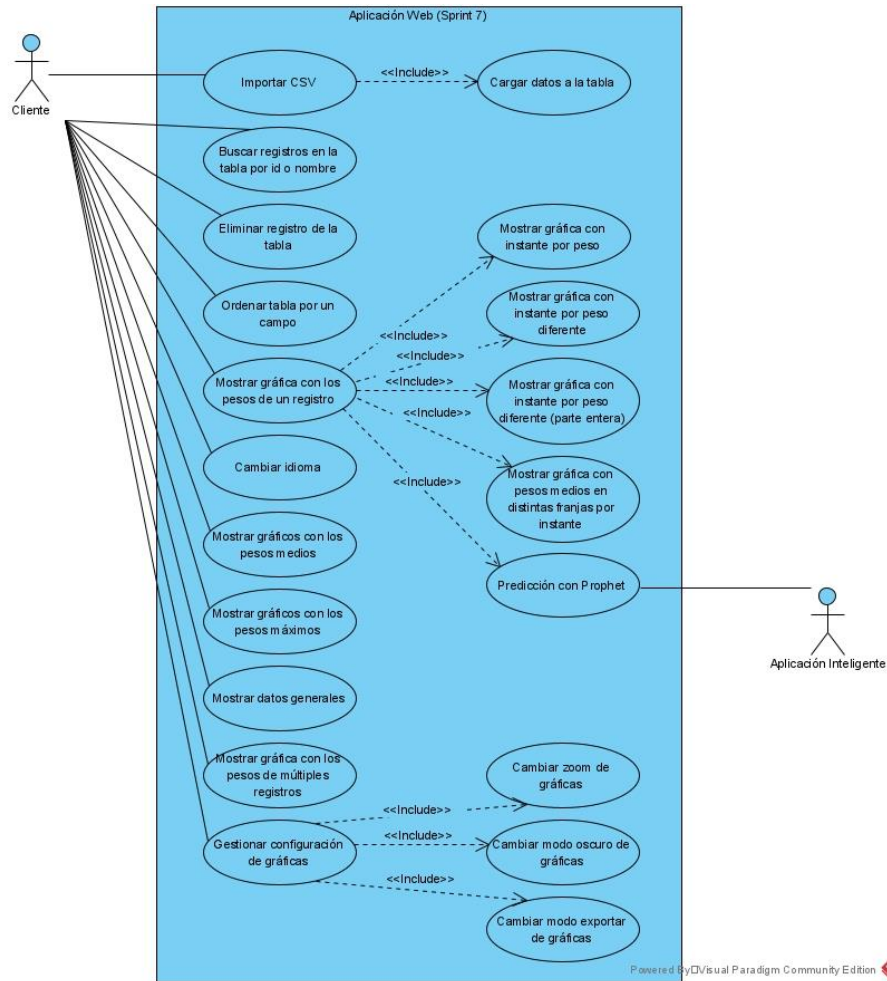


Ilustración 42. Diagrama casos de uso del Sprint 7

6.8.3. SPRINT 7 DISEÑO MEDIANTE UML

El siguiente diagrama UML de la ilustración 43, muestra la estructura Modelo-Vista-Controlador, junto a las funcionalidades implementadas en el Sprint 7 (únicamente se incluye el método *getGraficabyIdProphet* en el controlador del índice):

Aplicación web para la visualización de datos de sensor “Smart Bin”

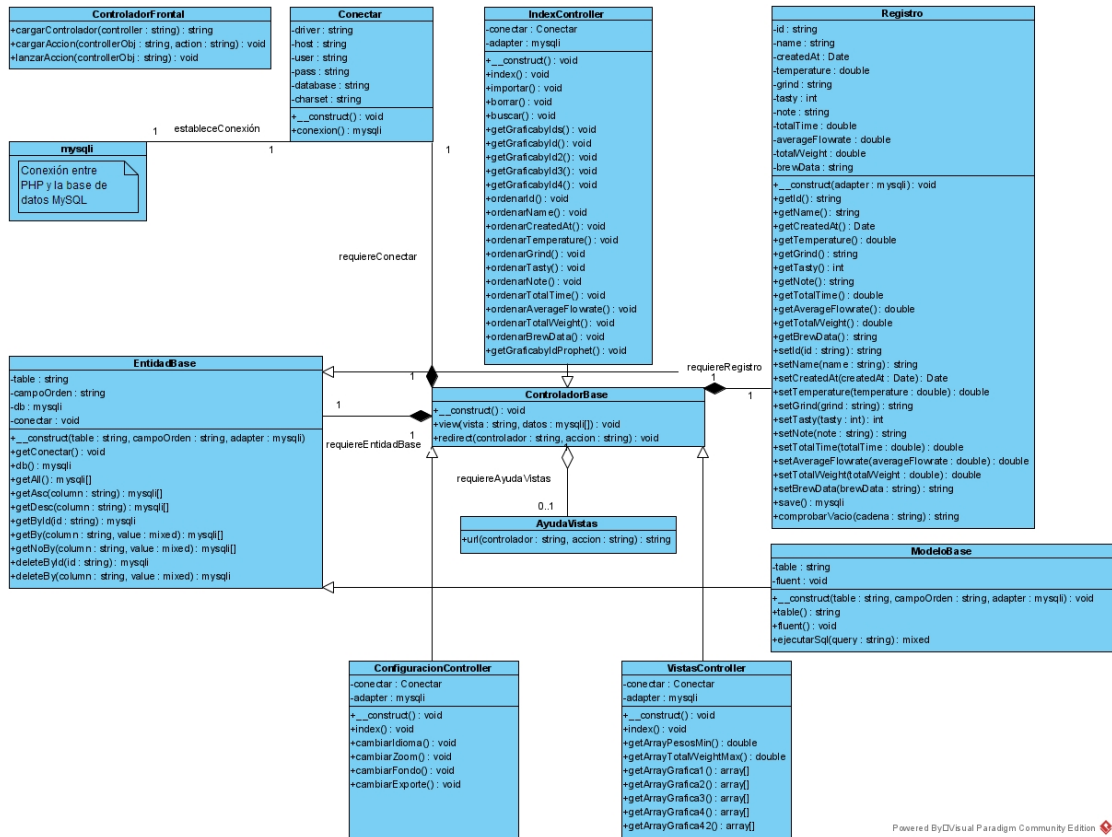


Ilustración 43. Diagrama UML del Sprint 7

6.8.4. SPRINT 7 DESARROLLO

En primer lugar, se organiza las tareas 18 y 19 (*Entorno de implementación para la integración y Conexión de ambas aplicaciones para la integración*):

- Descargar Anaconda Navigator.
- Crear entorno con las librerías necesarias para poder ejecutar localmente la aplicación inteligente.
- Ejecutar aplicación inteligente mediante el intérprete de comandos.

Se comienza descargando la aplicación Anaconda Navigator, que permite de forma gráfica y sin usar la línea de comandos, manejar paquetes, entornos y canales. Tras realizar la descarga, se modifica el entorno por defecto para que se pueda ejecutar la aplicación inteligente, siendo necesario importar las librerías:

- *python (3.8.5)*: necesario para ejecutar la aplicación inteligente.

- *scikit-learn* (0.24.1): necesario para realizar los cálculos matemáticos para generar las métricas.
- *pandas* (1.2.3): necesario para generar un *dataframe* (estructura de datos de dos dimensiones).
- *numpy* (1.19.2): también necesario para realizar cálculos matemáticos.
- *pystan* (2.19.1.1): necesario para realizar modelado y computación estadística con Python.
- *fbprophet* (0.7.1): necesario para usar Prophet.

Una vez preparado el entorno, se usa en la aplicación web la función *shell_exec*, que permite ejecutar un comando mediante el intérprete de comandos. Esto es necesario para activar el entorno de Anaconda (con el comando *conda activate base*) y ejecutar la aplicación inteligente.

Seguidamente, se organiza la tarea 20, 21 y 22 (*Datos de entrada para la integración, Modificación de la app inteligente para la integración y Datos de salida para la integración*):

- Generar arrays con los pesos que la aplicación web debe enviar a la aplicación inteligente.
- Modificar la aplicación inteligente para recibir los arrays con los pesos de la web, y poder enviar los arrays con los pesos predichos y las métricas.
- Generar pesos predichos y métricas que deben ser enviados a la aplicación web.

A diferencia de los arrays que se han estado manejando hasta ahora (con los pesos y los instantes de tiempo), en esta ocasión al tener que predecir, es más acertado trabajar con fechas. Por ello, los datos que va a mandar la aplicación web son los pesos medios por cada segundo, es decir, la misma estructura de datos que se había seleccionado como la más óptima en las pruebas de predicción de Prophet.

Aplicación web para la visualización de datos de sensor “Smart Bin”

Para permitir la transferencia de datos entre ambas aplicaciones, se realizaron ciertos cambios en la aplicación inteligente (*scriptProphet*) para leer y escribir ficheros de texto con los datos separados por “;”.

Por otro lado, los datos que debe enviar la aplicación inteligente a la web, son los pesos predichos con sus correspondientes fechas y las métricas resultantes: número de pesos iniciales, número de pesos predichos, R-Squared, MSE, MAE y RMSE.

Finalmente, se organiza la tarea 23 (*Generación de vistas para la integración*):

- Generar gráfico con los pesos iniciales y los predichos.
- Generar una tabla con las métricas.

Se crea un gráfico con dos series. Ambas con las fechas en su eje x y con el formato *Y-M-D H:M:S.MS*. Una de las series tiene en su eje y los pesos medios de entrada de los que se va a realizar la predicción (en gramos). Y la otra, tiene en su eje y los pesos predichos (también en gramos).

Por último, se crea una tabla con las métricas comentadas anteriormente.

6.8.5. SPRINT 7 REVIEW

Durante la reunión de revisión, en primer lugar se revisan los elementos del Sprint Backlog de la ilustración 44 que se han completado durante el desarrollo del séptimo Sprint:

| Incremento de Sprint 7: Historia de usuario 6: Integración de a... | Estado | Fecha Inicio | Fecha Fin | Esfuerzo esperado | Duración esperada | |
|--|--------|--------------|-----------|-------------------|-------------------|--|
| Tarea 18: Entorno de implementación para la integración | Listo | mar. 2 | mar. 3 | 8 hours | 10hours | |
| Tarea 19: Conexión de ambas aplicaciones para la integración | Listo | mar. 4 | mar. 7 | 16 hours | 20hours | |
| Tarea 20: Datos de entrada para la integración | Listo | mar. 8 | mar. 8 | 4 hours | 5hours | |
| Tarea 21: Modificación de la app inteligente para la integración | Listo | mar. 9 | mar. 9 | 4 hours | 5hours | |
| Tarea 22: Datos de salida para la integración | Listo | mar. 10 | mar. 10 | 4 hours | 5hours | |
| Tarea 23: Generación de vistas para la integración | Listo | mar. 11 | mar. 11 | 4 hours | 5hours | |
| + Agregar Activity | | | | | | |
| | | | | 40 hours Total | 50hours Total | |

Ilustración 44. Sprint 7 - Incremento

Seguidamente la reunión finaliza tras medir el progreso del proyecto mediante la gráfica de la ilustración 45:

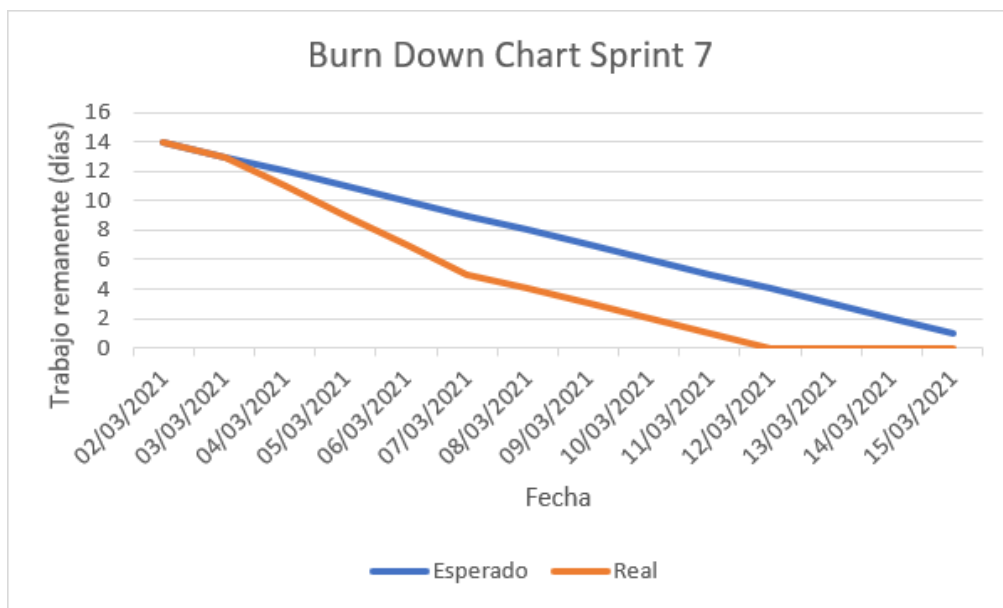


Ilustración 45. Burn Down Chart Sprint 7

6.8.6. SPRINT 7 RETROSPECTIVE

El enfoque inicial marcaba un desarrollo progresivo a lo largo de los días durante dos semanas. En cambio, el enfoque final demuestra una progresión más adelantada que la esperada, incluso se acaba antes de lo planeado el día 11/03/2021 (en el enfoque inicial se estimó que finalizaría el 15/03/2021). Ante la gran cantidad de tareas, se inició un trabajo más acelerado de lo habitual, lo que se ha traducido en esta finalización temprana.

6.9. SPRINT 8 - HISTORIA DE USUARIO 7: AJUSTE DE CURVA EN LA APLICACIÓN INTELIGENTE

6.9.1. SPRINT 8 PLANNING MEETING

Durante la reunión de planificación se establecen el Sprint Backlog y el Sprint Goal para el octavo Sprint en la tabla 66 e ilustración 46.

Además, como este Sprint no estaba previsto en la planificación inicial, también se define una nueva historia de usuario (*Historia de usuario 7: Ajuste de curva en la aplicación inteligente*) en la tabla 60 y varias tareas nuevas que se muestran de la tabla 61 a la 65.

| Historia de usuario 7: Ajuste de curva en la aplicación inteligente | |
|---|---------|
| Rol | Cliente |

| | |
|--------------------------------|--|
| Necesidad y beneficio | Crear una aplicación inteligente que incluya un ajuste de curva en los pesos iniciales que se toman como referencia para la predicción de Prophet. |
| Criterio de aceptación (SMART) | La aplicación inteligente incluye una variante con ajuste de curva y esta también se integra en la aplicación web. |

Tabla 60. Generación de predicciones

| | | |
|---|------------------------|--|
| Nombre del integrante: Alberto Pons Buttazzo | | |
| Descripción de la actividad: generar datos de entrada que la aplicación web debe enviar a la aplicación inteligente con ajuste de curva. | | |
| Duración: 1 día | Esfuerzo: 1 día | Necesario para historias de usuario: 7 |

Tabla 61. Tarea 24: Datos de entrada para la integración con ajuste de curva

| | | |
|--|-------------------------|--|
| Nombre del integrante: Alberto Pons Buttazzo | | |
| Descripción de la actividad: modificar aplicación inteligente para que realice un ajuste de curva sobre los pesos de entrada. | | |
| Duración: 3 días | Esfuerzo: 3 días | Necesario para historias de usuario: 7 |

Tabla 62. Tarea 25: Modificación de la app inteligente con ajuste de curva

| | | |
|--|------------------------|--|
| Nombre del integrante: Alberto Pons Buttazzo | | |
| Descripción de la actividad: generar datos de salida que la aplicación inteligente con ajuste de curva debe enviar a la aplicación web. | | |
| Duración: 1 día | Esfuerzo: 1 día | Necesario para historias de usuario: 7 |

Tabla 63. Tarea 26: Datos de salida para la integración con ajuste de curva

| |
|---|
| Nombre del integrante: Alberto Pons Buttazzo |
|---|

| | | |
|---|------------------------|--|
| Descripción de la actividad: recibir datos generados por la aplicación inteligente con ajuste de curva y generar vistas para mostrarlos. | | |
| Duración: 1 día | Esfuerzo: 1 día | Necesario para historias de usuario: 7 |

Tabla 64. Tarea 27: Generación de vistas para la integración con ajuste de curva

| | | |
|---|------------------------|--|
| Nombre del integrante: Alberto Pons Buttazzo | | |
| Descripción de la actividad: ejecutar spinner mientras carga alguna de las dos variantes de la aplicación inteligente. | | |
| Duración: 1 día | Esfuerzo: 1 día | Necesario para historias de usuario: 6 y 7 |

Tabla 65. Tarea 28: Carga con spinner para la aplicación inteligente

| SPRINT BACKLOG | |
|--|--|
| Tareas | Metodologías |
| Tareas 24, 25, 26 y 27. | Metodologías 24, 25, 26 y 27: Similares a las de las tareas 20, 21, 22 y 23, pero tratando el ajuste de curva. |
| Tarea 28. | Metodología 28: Mostrar un spinner de carga al ejecutar los formularios de las dos variantes de la aplicación inteligente. |
| SPRINT GOAL | |
| El objetivo del Sprint 8 es generar un nuevo tipo de predicción con Prophet, tomando como referencia los pesos con un ajuste de curva. | |

Tabla 66. Sprint 8 Backlog

| Estado | Fecha Inicio | Fecha Fin | Esfuerzo esperado | Duración esperada |
|--------------------|--------------|-----------|-------------------|-------------------|
| En espera | mar. 16 | mar. 16 | 4 hours | 5hours |
| En espera | mar. 17 | mar. 19 | 12 hours | 15hours |
| En espera | mar. 20 | mar. 20 | 4 hours | 5hours |
| En espera | mar. 21 | mar. 21 | 4 hours | 5hours |
| En espera | mar. 22 | mar. 22 | 4 hours | 5hours |
| + Agregar Activity | | | | |
| | | | 28 hours Total | 35hours Total |

Ilustración 46. Sprint 8 - Sprint Backlog inicial

6.9.2. SPRINT 8 ANÁLISIS MEDIANTE CASOS DE USO

El siguiente diagrama de casos de uso de la ilustración 47, muestra las funcionalidades que se van a implementar en la web en el Sprint 8:

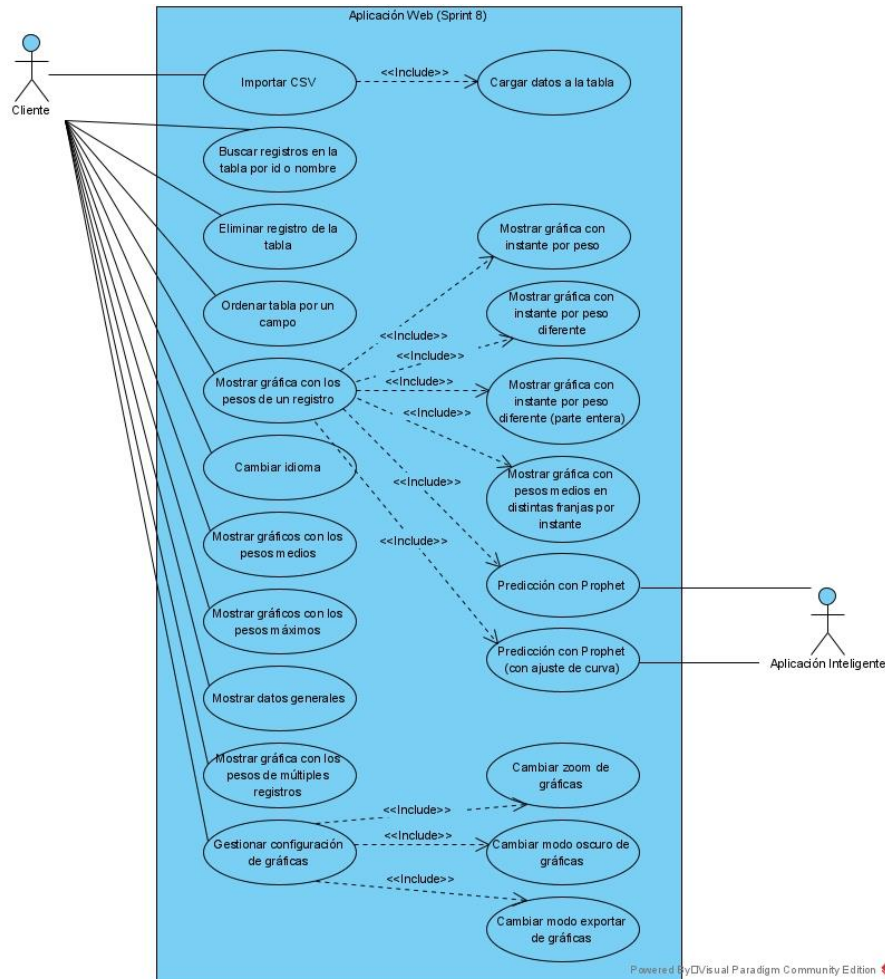


Ilustración 47. Diagrama casos de uso del Sprint 8

6.9.3. SPRINT 8 DISEÑO MEDIANTE UML

El siguiente diagrama UML de la ilustración 48, muestra la estructura Modelo-Vista-Controlador, junto a las funcionalidades implementadas en el Sprint 8 (únicamente se incluye el método *getGraficabyIdProphetFit* en el controlador del índice):

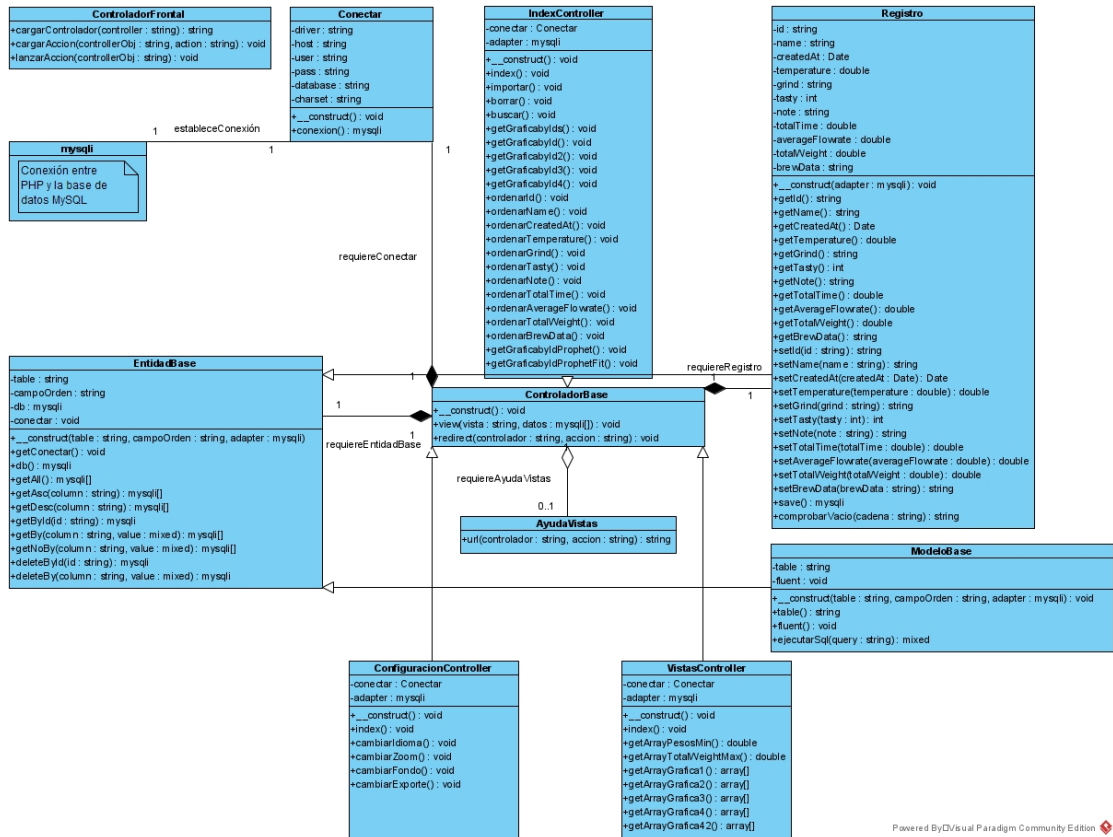


Ilustración 48. Diagrama UML del Sprint 8

6.9.4. SPRINT 8 DESARROLLO

En primer lugar, se organiza las tareas 24, 25 y 26 (*Datos de entrada para la integración con ajuste de curva, Modificación de la app inteligente para la integración con ajuste de curva y Datos de salida para la integración con ajuste de curva*):

- Generar arrays con los pesos que la aplicación web debe enviar a la aplicación inteligente.
- Crear una variante de la aplicación inteligente, pero usando un ajuste de curva sobre los pesos de entrada que se reciben de la web.
- Generar pesos predichos y métricas que deben ser enviados a la aplicación web.

A diferencia de la aplicación inteligente original, los pesos de entrada no son los pesos medios en franjas, sino la totalidad de los pesos. No se usa la otra variante de pesos, ya que al usar un ajuste de curva, ya existe un método para

suavizar la curva. Por ello, los datos que va usar como entrada el algoritmo de Prophet, son los pesos con una ajuste de curva que vienen definidos por una ecuación polinómica de segundo grado: $y = ax + bx^2 + c$.

Cuanto mayor sea el grado de la ecuación, mayor es la curva que se genera. Esto se puede ver reflejado en la ilustración 49 (Wikipedia, 2021), siendo la curva verde, el tipo de curva que se va a usar.

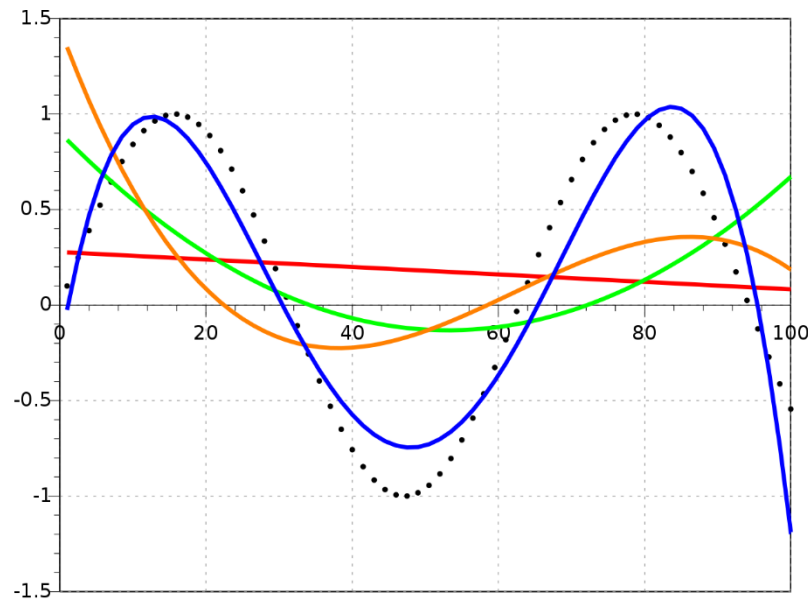


Ilustración 49. Tipos de curvas

Los datos que debe enviar la aplicación inteligente (*scriptProphetFit*) a la web, son los mismos que en la variante original, pero incluyendo los pesos con ajuste de curva.

A continuación, se organiza la tarea 27 (*Generación de vistas para la integración con ajuste de curva*):

- Generar gráfico con los pesos iniciales, los pesos con el ajuste de curva y los pesos predichos.
- Generar una tabla con las métricas.

Se crea un gráfico con tres series. Todas con las fechas en su eje x y con el formato *Y-M-D H:M:S.MS*. Una de las series tiene en su eje y los pesos iniciales, otra los pesos con el ajuste de curva, y la última los pesos predichos (todos en gramos).

También se crea una tabla con las métricas, al igual que con la otra variante de la aplicación inteligente. Dichas métricas reflejan errores muy pequeños y un porcentaje de variación que el modelo es capaz de explicar (R-Squared) que en todos los casos es superior al 99,9% (0,999).

Finalmente, se organiza la tarea 28 (*Carga con spinner para la aplicación inteligente*):

- Mostrar un spinner de carga al ejecutar los formularios de las dos variantes de la aplicación inteligente.

Debido a que las aplicaciones inteligentes tardan en cargar entre 10 y 45 segundos, es necesario mostrar un spinner de carga para que la persona tenga constancia que se está ejecutando el algoritmo. El tiempo de carga puede variar por el número de pesos de entrada que recibe el algoritmo y por el procesador que tenga el ordenador.

6.9.5. SPRINT 8 REVIEW

Durante la reunión de revisión, en primer lugar se revisan los elementos del Sprint Backlog de la ilustración 50 que se han completado durante el desarrollo del octavo Sprint:

| Incremento de Sprint 8: Historia de usuario 7: Ajuste de curva e... | Estado | Fecha Inicio | Fecha Fin | Esfuerzo real | Duración real | |
|--|--------|--------------|--------------|-------------------|------------------|--|
| Tarea 24: Datos de entrada para la integración con ajuste de curva | Listo | mar. 16 | mar. 16 | 8 hours | 10hours | |
| Tarea 25: Modificación de la app inteligente con ajuste de curva | Listo | mar. 17 | mar. 19 | 20 hours | 25hours | |
| Tarea 26: Datos de salida para la integración con ajuste de curva | Listo | mar. 20 | mar. 20 | 8 hours | 10hours | |
| Tarea 27: Generación de vistas para la integración con ajuste de curva | Listo | mar. 21 | mar. 21 | 8 hours | 10hours | |
| Tarea 28: Carga con spinner para la aplicación inteligente | Listo | mar. 22 | mar. 22 | 4 hours | 5hours | |
| + Agregar Activity | | | | | | |
| | | mar. 16 - 22 | mar. 16 - 22 | 48 hours Total | 60hours Total | |

Ilustración 50. Sprint 8 - Incremento

Seguidamente la reunión finaliza tras medir el progreso del proyecto mediante la gráfica de la ilustración 51:

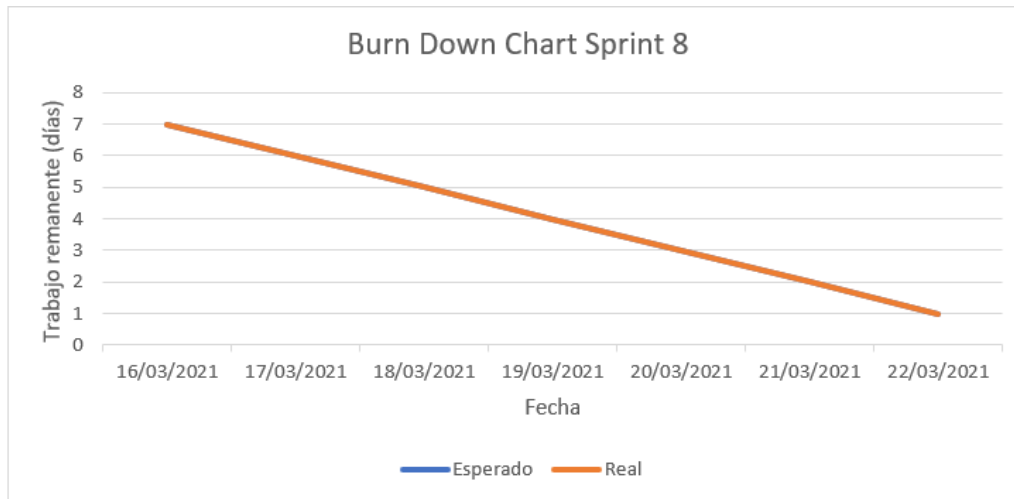


Ilustración 51. Burn Down Chart Sprint 8

6.9.6. SPRINT 8 RETROSPECTIVE

En este caso, siendo el último Sprint, se ha seguido de forma progresiva la ejecución de las tareas tal y como se estimó. Esto se debe a que esta planificación se realizó nada más acabar el anterior Sprint. Además de que el Sprint es de 7 días en vez de 14, con lo que es más fácil seguir la planificación. Con esta última retrospectiva, finaliza por completo el desarrollo de la aplicación inteligente y la aplicación web.

6.10. RETROSPECTIVA GENERAL DE TODO EL DESARROLLO

Finalmente, para acabar con el desarrollo, se resume el mismo con una comparativa entre la planificación inicial y lo que realmente se ha acabado elaborando. Esto se refleja con una comparativa de los dos diagramas de las ilustraciones 52 y 53, y con un Burndown de todo el desarrollo en la ilustración 54:

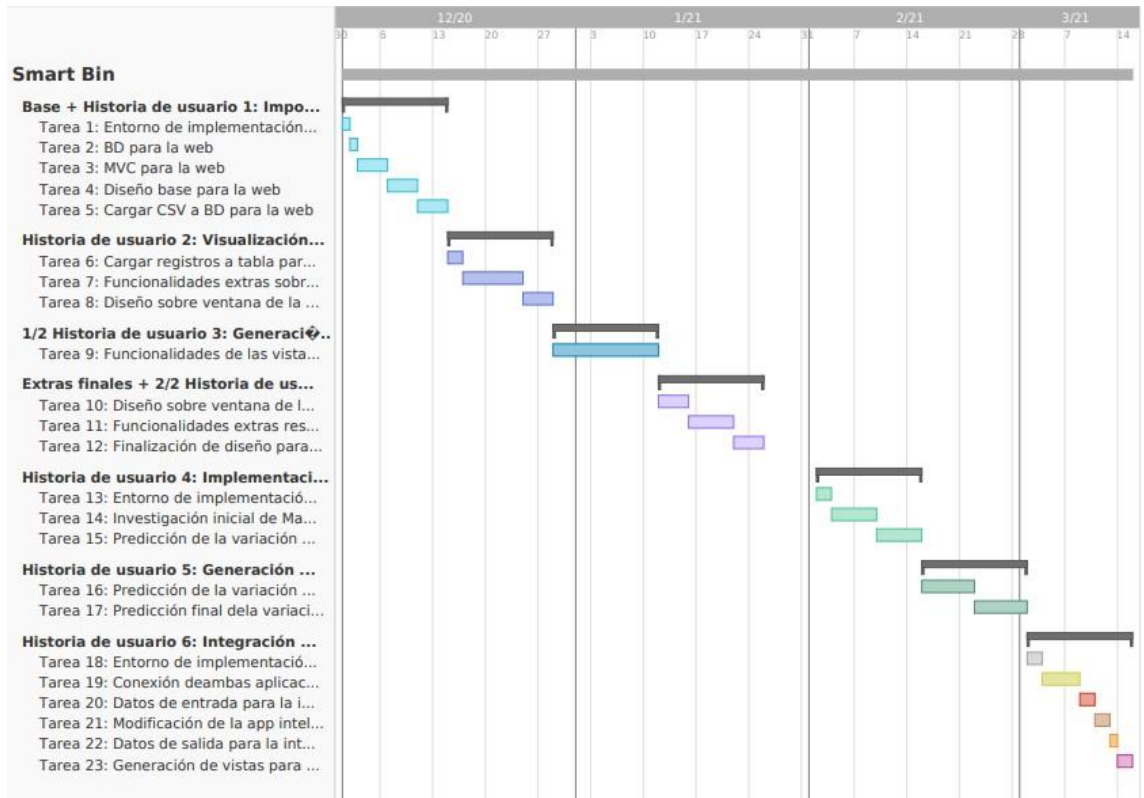


Ilustración 52. Gantt inicial

Aplicación web para la visualización de datos de sensor "Smart Bin"

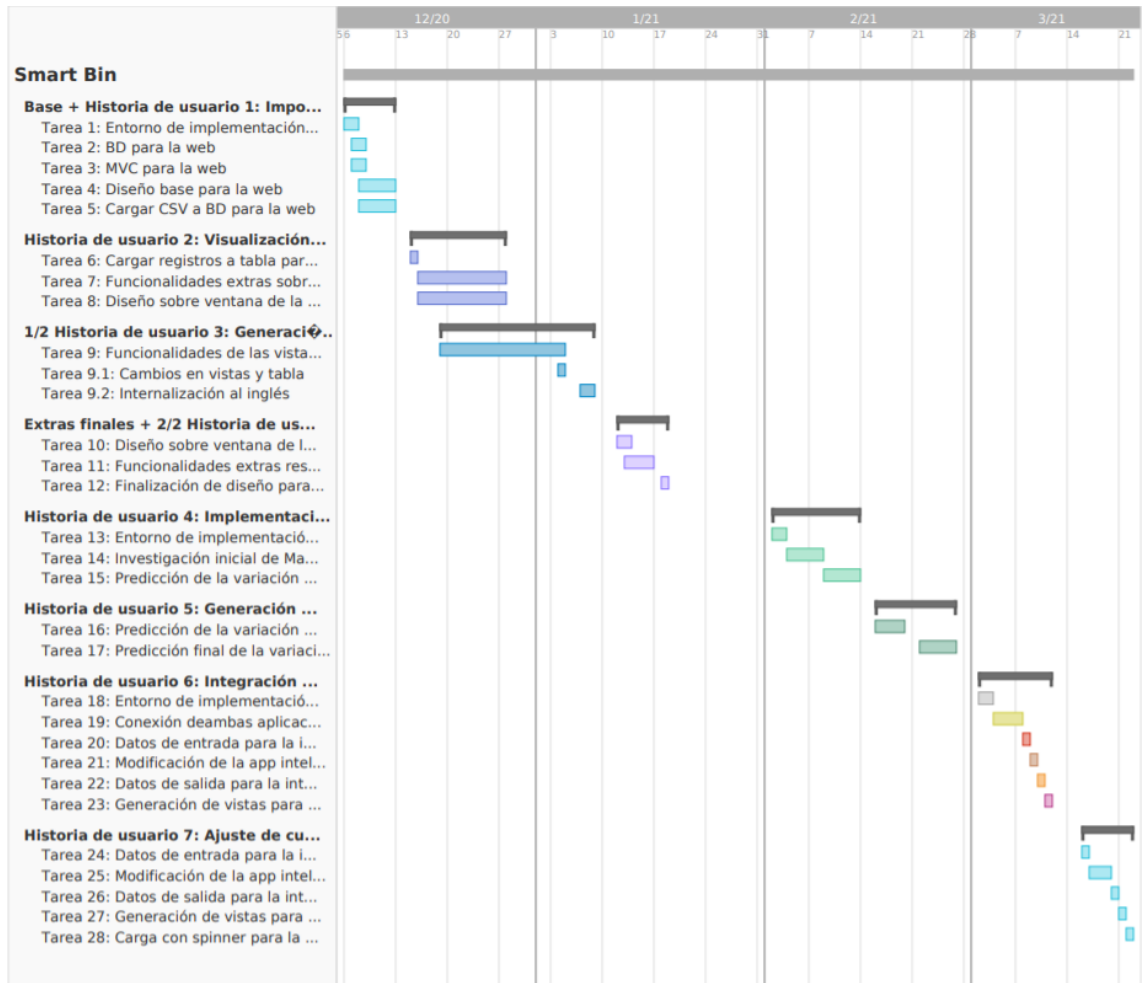


Ilustración 53. Gantt final

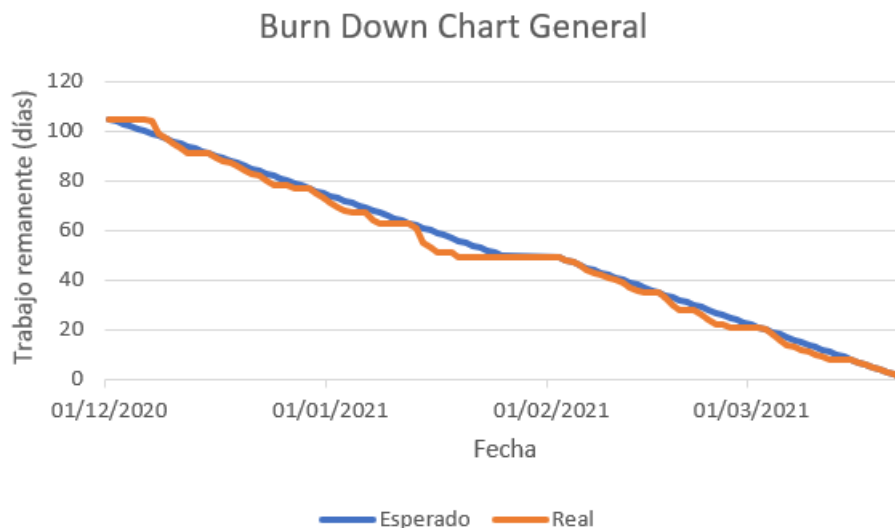


Ilustración 54. Burn Down Chart General

Se puede observar que aunque existen desviaciones de días entre las comparativas, por lo general se ha seguido lo que se planificó inicialmente.

Únicamente cabe destacar el último Sprint, el cual no estaba reflejado en la planificación inicial. Con todo ello, el proyecto se inició 6 días más tarde de lo esperado, es decir que en vez de empezar el 01/12/2020, se inició el 06/12/2020. Por otro lado, en vez de acabar el 15/03/2021, se acabó 7 días más tarde, el 22/03/2021.

Finalmente, el desarrollo ha resultado en: 8 Sprints, 7 historias de usuario, 28 tareas y 390 horas de trabajo, resultando en una media de 14 horas dedicadas a cada tarea.

7. DESPLIEGUE Y PRUEBA DE LA SOLUCIÓN

7.1. PLAN DE PRUEBAS

Cuando se trabaja con Scrum, se deben realizar pruebas al final de cada Sprint. Como se comentó en el Sprint 1, se han usado dos herramientas para revisar el código a lo largo del proyecto: (1) PHP Intelephense para seguir los estándares de PHP y (2) SonarLint para revisar errores en el código.

Estas herramientas permiten detectar errores y evita que se tenga código “sucio”. Por ejemplo, estas herramientas detectan el uso de un mismo nombre en distintas variables, métodos que necesitan ser refactorizados por tener demasiada complejidad, el exceso en el número de retornos permitidos, cuando se usa un tipo de dato incorrecto, etc. Esto permite probar el código de forma semiautomática, es decir, los errores se detectan automáticamente, aunque la corrección sea manual o mediante las sugerencias que ofrecen las herramientas para corregir el error.

Además de estas herramientas, se ha usado el plan de pruebas TDD (*Test-driven development*) (Wikipedia, 2020), que suele ser usado en desarrollo del software ágil. TDD implica el uso de dos prácticas: (1) definir las pruebas primero (por lo general pruebas unitarias) y (2) refactorizar, es decir, reestructurar y cambiar el código sin alterar su comportamiento. Para esto último, se han usado las herramientas de apoyo que se comentaron anteriormente, con el propósito de tener un código “limpio”.

A continuación, se van a mostrar las pruebas unitarias que se han realizado sobre las principales funciones del software. Por otro lado, se van a ir respondiendo las 5 preguntas que toda prueba unitaria debe contestar (Elliott, 2015): (1) ¿Qué se prueba? (2) ¿Qué debería hacer? (3) ¿Cuál es la salida actual? (4) ¿Cuál es la salida esperada? (5) ¿Cómo se puede reproducir la prueba? La última pregunta es respondida por el código utilizado para derivar del valor “real” al valor “esperado”. Esta se responde cuando la prueba no ha ofrecido la salida esperada.

Para un mayor seguimiento y claridad de lo que se prueba, se va a usar como referencia el diagrama de casos de uso de la ilustración 55 que se ha ido

desarrollando durante la documentación. Por otro lado, se reflejan cada una de las pruebas unitarias en la tabla 67.

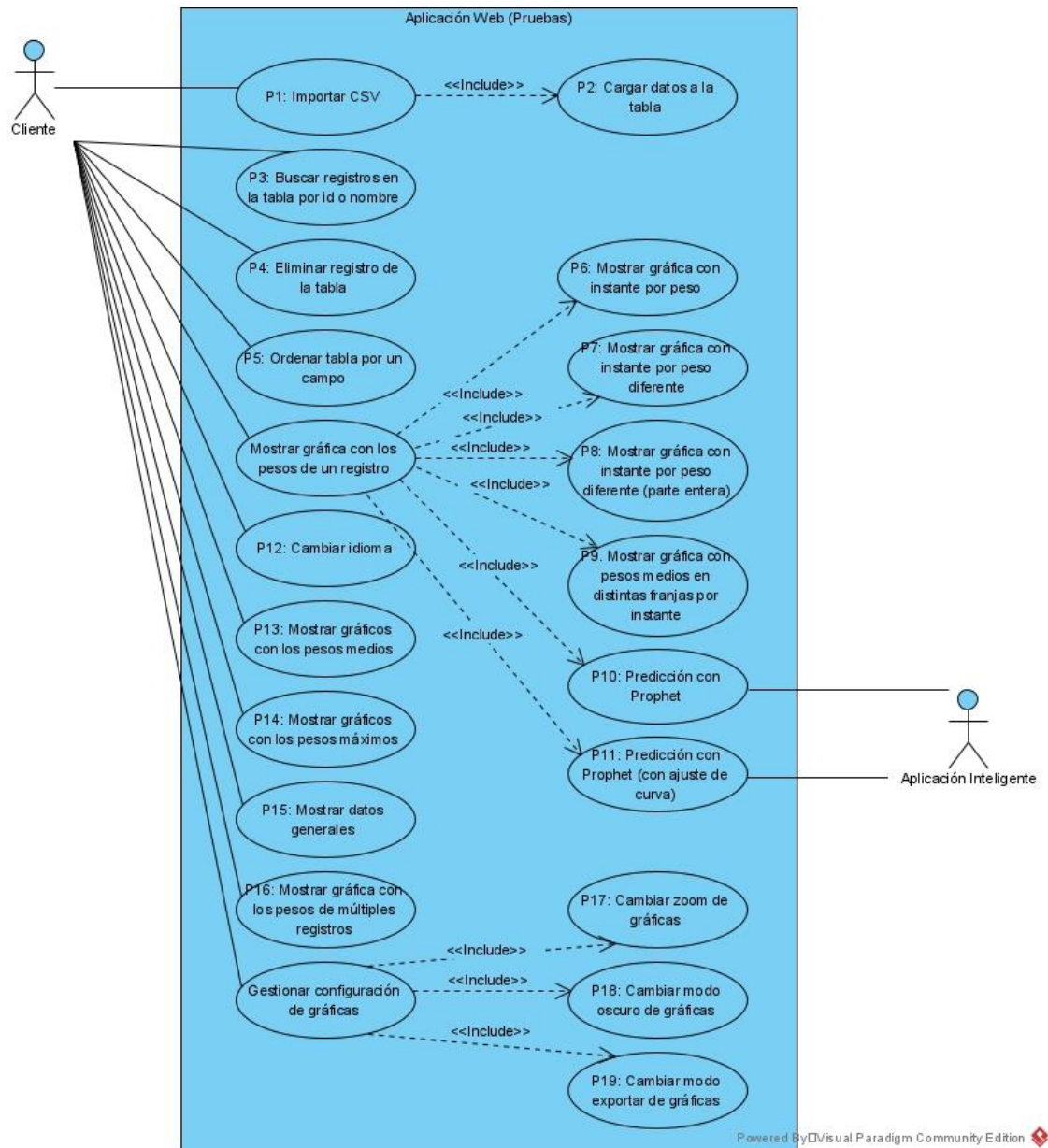


Ilustración 55. Diagrama casos de uso con pruebas

| Nombre de la prueba | ¿Qué se prueba? | ¿Qué debería hacer? | ¿Cuál es la salida esperada? | ¿Cuál es la salida actual? | ¿Cómo se puede reproducir la prueba? |
|---------------------|-----------------------------|---|---------------------------------|----------------------------|--------------------------------------|
| P1: Importar CSV | La importación de registros | Recorrer el CSV y almacenar los registros en la | BD con los registros importados | “” | - |

Aplicación web para la visualización de datos de sensor “Smart Bin”

| | | | | | |
|--|--|---|---|---|---|
| | de un CSV a la BD | BD mediante consultas INSERT | | | |
| P2: Cargar datos a la tabla | La carga de los datos de la BD a la tabla | Recorrer todos los registros de la BD y que la tabla los muestre | Tabla con todos los registros de la BD | Tabla con los datos esparcidos en columnas y filas que no le corresponde n. Esto ocurre cuando habían columnas con cadenas muy largas | Aumentar el tamaño del parámetro <i>length</i> de la función <i>fgetcsv()</i> a 50000 |
| P3: Buscar registro en la tabla por id o nombre | La búsqueda de los registros de la tabla por su id o nombre | Filtrar la consulta de búsqueda por el id o el nombre (id LIKE <i>chain</i> OR name LIKE <i>chain</i>) | Tabla con el id o el nombre propuesto. | “ | - |
| P4: Eliminar registro de la tabla | La eliminación de un registro de la tabla, y por ende de la BD | Eliminar registro de la BD por su id mediante una consulta DELETE y volver a recorrer la tabla | Tabla con todos los registros, menos con el registro que se acaba de eliminar | “” | - |
| P5: Ordenar tabla por un campo | La ordenación de la tabla de forma ascendente en base al | Ordenar los registros consultados de forma ascendente en base al campo | Tabla ordenada ascendentemente en base al campo seleccionado | “” | - |

| | campo seleccionado | seleccionado (ORDER BY <i>column</i> ASC) | | | |
|--|---|---|--|---|---|
| P6: Mostrar gráfica con instante por peso | La integridad de los datos que la gráfica muestra | Generar un array con los pesos y sus instantes correspondient es; se vuelca todo en la gráfica | Gráfica con instante por peso | " | - |
| P7: Mostrar gráfica con instante por peso diferente | La integridad de los datos que la gráfica muestra | Generar un array con los pesos diferentes y sus instantes correspondient es; se vuelca todo en la gráfica | Gráfica con instante por peso diferente | " | - |
| P8: Mostrar gráfica con instante por peso diferente (parte entera) | La integridad de los datos que la gráfica muestra | Generar un array con los pesos diferentes (parte entera) y sus instantes correspondient es; se vuelca todo en la gráfica | Gráfica con instante por peso diferente (parte entera) | " | - |
| P9: Mostrar gráfica con pesos medios en distintas franjas | La integridad de los datos que la gráfica muestra | Generar un array con los pesos medios y sus instantes correspondient es; se vuelca todo en la gráfica | Gráfica con pesos medios en distintas franjas por instante | " | - |

Aplicación web para la visualización de datos de sensor “Smart Bin”

| | | | | | |
|--|---|--|--|--|---|
| por instante | | | | | |
| P10: Predicción con Prophet | La integridad de los datos que la gráfica muestra | Generar (1) un array con los pesos medios y sus fechas correspondientes, (2) un array con los pesos predichos y sus fechas correspondientes, (3) y un array con las métricas de la aplicación inteligente; se vuelca todo en la gráfica | Gráfica con (1) pesos medios en distintas franjas por fecha, (2) gráfica con la predicción de Prophet, (3) y tabla con métricas | Error de integración cuando se pasan los arrays por parámetro a la aplicación inteligente: <i>Warning:</i> <i>exec():</i> <i>Unable to fork</i> | Pasar arrays por ficheros de texto en vez de por parámetros |
| P11: Predicción con Prophet (con ajuste de curva) | La integridad de los datos que la gráfica muestra | Generar (1) un array con los pesos y sus fechas correspondientes, (2) un array con los pesos y sus fechas correspondientes (con ajuste de curva), (3) un array con los pesos predichos y sus fechas correspondientes, (4) y un array con las métricas de la aplicación inteligente; se | Gráfica con (1) pesos por fecha, (2) gráfica con el ajuste de curva,(3) gráfica con la predicción de Prophet, (4) y tabla con métricas | “ | - |

| | | | | | |
|--|--|---|--|--|---|
| | | vuelca todo en la gráfica | | | |
| P12: Cambiar idioma | El cambio de idioma de la web de castellano a inglés y viceversa | Cambiar entre un fichero con las cadenas de texto en castellano al otro con las cadenas de texto en inglés, y viceversa | Web con todas las cadenas de texto en el idioma seleccionado | Error cuando se hace referencia a una llave que no existe en el array de cadenas: <i>Undefined array key</i> | Incluir traducción en ambos ficheros de texto (castellano o inglés) o renombrar la clave del array con las traducciones |
| P13: Mostrar gráficos con los pesos medios | La integridad de los datos que la gráfica muestra | Generar (1) un array con los pesos medios y sus registros correspondientes, (2) y un array con los pesos medios y sus fechas correspondientes; se vuelca todo en la gráfica | Gráficas con los pesos medios | "" | - |
| P14: Mostrar gráficos con los pesos máximos | La integridad de los datos que la gráfica muestra | Generar (1) un array con los pesos máximos y sus registros correspondientes, (2) y un array con los pesos máximos y sus fechas correspondientes; se vuelca todo en la gráfica | Gráficas con los pesos máximos | "" | - |

Aplicación web para la visualización de datos de sensor “Smart Bin”

| | | | | | |
|--|---|--|--|--|---|
| P15: Mostrar datos generales | La integridad de los datos que se muestran | Generar (1) el número de registros totales, (2) el número de los registros sin pesos, (3) el peso más alto, (4) y el peso más bajo; se vuelca todo en la vista | Datos con (1) el número de registros totales, (2) el número de los registros sin pesos, (3) el peso más alto, (4) y el peso más bajo | Error cuando no hay registros en la tabla: <i>Uncaught ValueError</i> | Establecer datos a un valor predeterminado ante un fallo en la consulta (0 cuando no hay registros) |
| P16: Mostrar gráfica con los pesos en múltiples registros | La integridad de los datos que la gráfica muestra | Generar un array con los pesos y sus instantes correspondientes de múltiples registros; se vuelca todo en la gráfica | Gráfica con los pesos en múltiples registros | “” | - |
| P17: Cambiar zoom de gráficas | La habilitación o deshabilitación del zoom | Cambiar parámetro <i>zoomEnabled</i> de <i>true</i> a <i>false</i> , y viceversa | Las gráficas permiten o no permitir hacer zoom, en base a la selección realizada | “” | - |
| P18: Cambiar modo oscuro de gráficas | La habilitación o deshabilitación del modo oscuro | Cambiar parámetro <i>theme</i> de <i>light</i> a <i>dark</i> , y viceversa | Las gráficas activan o desactivan el modo oscuro, en base a la selección realizada | “” | - |
| P19: Cambiar modo exportar de gráficas | La habilitación o deshabilitación del modo exportar | Cambiar parámetro <i>exportEnabled</i> de <i>true</i> a <i>false</i> , y viceversa | Las gráficas activan o desactivan el modo exportar, en base a la | “” | - |

| | | | | | |
|--|--|--|------------------------|--|--|
| | | | selección realizada | | |
|--|--|--|------------------------|--|--|

Tabla 67. Pruebas unitarias

7.2. INDICADORES DEL DESEMPEÑO Y/O UTILIDAD

Para determinar el rendimiento del trabajo realizado se van a usar cuatro métricas KPI (indicadores claves del rendimiento) (Bancroft-Connors, 2020):

- *Cycle Time*: indica la productividad durante un Sprint (diferencia entre el inicio y el final de un Sprint).
- *Escaped Defect Rate*: indica la calidad del trabajo realizado durante un Sprint (número de defectos). Un defecto en este caso se considera una imperfección que se identifica después de que acabe un Sprint, es decir, un error sobre una tarea que se realizó en un Sprint anterior.
- *Planned-to-Done Ratio*: indica la predictibilidad que ha habido en un Sprint (porcentaje de trabajo realizado según lo planeado).
- *Happiness Metric*: refleja la estabilidad del equipo de trabajo al final de un Sprint (moral del equipo baja, media o alta).

A continuación, en la tabla 58 se pueden observar cada una de las métricas que se expusieron anteriormente por cada Sprint.

| | Cycle Time (Media en días) | Escaped Defect Rate | Planned to Done Ratio (%) | Happiness Metric |
|-----------------|---------------------------------------|----------------------------|----------------------------------|-------------------------|
| Sprint 1 | 7 | 0 | 100 | alta |
| Sprint 2 | 13 | 1 | 100 | media |
| Sprint 3 | 21 | 2 | 100 | media |
| Sprint 4 | 7 | 0 | 100 | media |
| Sprint 5 | 12 | 0 | 100 | media |
| Sprint 6 | 11 | 0 | 100 | media |
| Sprint 7 | 10 | 1 | 100 | media |
| Sprint 8 | 7 | 0 | 100 | alta |

| | | | | |
|--------------|----|------------------------------|-----|-------|
| Media | 11 | 1/2 (1 cada 2 Sprints) | 100 | media |
|--------------|----|------------------------------|-----|-------|

Tabla 68. Rendimiento mediante valores KPI

7.3. ESCALABILIDAD Y MANTENIMIENTO

La escalabilidad y el mantenimiento de la aplicación fueron previstos desde el principio del desarrollo, por ello se ha seguido un patrón MVC. Dos de las grandes ventajas de este patrón son la facilidad en el manejo de errores y la gran escalabilidad que ofrece (Gómez, 2015). Ya que, al caracterizarse por una gran modularización, es posible detectar más rápido los errores, y se permite incluir nuevos módulos y funciones con mayor facilidad.

Por otro lado, todo el código viene acompañado con múltiples comentarios, explicando métodos, atributos, clases, etc. Esta decisión fue tomada para que cualquier persona que pueda tener alguna duda sobre el código, pueda resolverla con mayor facilidad. Además dichos comentarios están disponibles tanto en castellano como en inglés.

Finalmente, las posibles nuevas funcionalidades y mejoras que el programa puede incluir se reflejan en el siguiente apartado 8, concretamente en el apartado 8.3 sobre las vías futuras del proyecto.

8. CONCLUSIONES

8.1. OBJETIVOS ALCANZADOS

Se van a exponer los objetivos que se han cumplido mediante la tabla 59. En esta tabla se distinguen cinco columnas: la primera especifica el nombre del objetivo, la segunda describe el objetivo, la tercera señala si el objetivo se ha podido cumplir, la cuarta indica cómo se ha cumplido dicho objetivo y la última columna especifica si el objetivo fue previsto inicialmente. Los dos primeros objetivos hacen referencia a la creación de la aplicación web, mientras que los dos siguientes se relacionan con la creación de la aplicación inteligente.

| Nombre | Descripción | ¿Se cumple? | ¿Cómo se ha cumplido? | ¿Fue previsto inicialmente? |
|---------------------------------------|--|--------------------|---|------------------------------------|
| Procesamiento de ficheros CSV (OE1.1) | Procesar ficheros en formato CSV y contener un repositorio con los valores que se obtienen de los distintos experimentos que se realicen | Sí | Se leen cada una de las filas de un fichero CSV con los registros de los experimentos, y estas se almacenan en una base de datos mediante un módulo MySQL de phpMyAdmin | Sí |
| Generación de vistas (OE1.2) | Generar diferentes vistas (gráficos, tablas, etc.) de los datos recogidos para ofrecer información relevante de los mismos | Sí | Los datos registrados en la base de datos se recuperan, y se ven reflejados mediante una tabla y varios gráficos mediante los scripts de la | Sí |

| | | | página canvasJS | |
|---|---|----|---|----|
| Evaluación de algoritmos fiables (OE2.1) | Evaluar un conjunto de algoritmos que permitan encontrar patrones de variación de peso con fiabilidad | Sí | Se estudian las librerías Prophet y Arima, y se usan sobre el conjunto de datos de los experimentos | Sí |
| Generación de predicciones precisas (OE2.2) | Hacer uso de los algoritmos evaluados en OE2.1 para asegurar un gran porcentaje de precisión en las futuras variaciones de los desechos | Sí | Se optimiza la predicción del algoritmo Prophet (en la mayoría de casos predicciones superiores al 80% de precisión) mediante la modificación de los dataframes que el algoritmo va a usar y mediante la modificación de tres de sus parámetros | Sí |
| Integración de ambas aplicaciones (OE3) | Tanto la aplicación web, como la inteligente, se pueden usar desde una única aplicación | Sí | La aplicación web se comunica con la aplicación inteligente mediante la función <i>shell_exec</i> (ejecuta un comando mediante el | No |

| | | | | |
|--|--|--|---|--|
| | | | intérprete de comandos) y mediante el intercambio de ficheros <code>.txt</code> | |
|--|--|--|---|--|

Tabla 69. Cumplimiento de objetivos

8.2. CONCLUSIONES DEL TRABAJO

En primer lugar, sobre el trabajo realizado se puede concluir mediante la tabla del punto 8.1, que se ha conseguido cumplir todos los objetivos que se habían propuesto inicialmente. Los objetivos alcanzados se traducen en los siguientes resultados:

- Repositorio que almacena los datos de los experimentos de un sensor de peso.
- Visualización de dichos datos mediante una tabla y varios gráficos con: pesos medios, pesos máximos, todos los pesos, pesos alterados suavizando la curva, pesos predichos, etc.
- Predicción de los pesos de un experimento mediante Prophet, obteniendo errores relativamente pequeños y con predicciones con una precisión superior al 80% en la mayor parte de casos (del 99,9% aplicando un ajuste de curva).
- Otros: internalización de la web, configuración de vistas, filtro de datos, etc.

En segundo lugar, sobre las elecciones tomadas durante el trabajo, se puede concluir que la lesividad, adaptación y seguimiento que ofrece la metodología Scrum ha permitido que los resultados obtenidos hayan sido satisfactorios por parte del cliente. Por un lado, gracias a la continua comunicación con el cliente se ha conseguido el cumplimiento de los objetivos cómo el cliente lo deseaba. Por otro lado, gracias a la capacidad adaptativa y a la planificación que desde un punto de vista ágil puede ser incluso excesiva, se ha conseguido terminar el proyecto solamente con una semana de retraso.

Finalmente, respondiendo a la pregunta: Si se volviera a realizar el trabajo ¿Se haría igual? ¿Qué se ha aprendido? Esta pregunta se va a responder partiendo de la experiencia que se ha obtenido durante el desarrollo y en base al mayor entendimiento que se tiene sobre las metodologías ágiles. Por un lado, se definirían puntos de historia con la misma duración, simplificando así la organización de cada Sprint. Por otro lado, se usaría la experiencia obtenida para definir con mayor precisión la magnitud de cada tarea. Por último, se elaboraría un sistema de despliegue desde el principio del proyecto que permitiese que el cliente pudiese probar la aplicación sin tener que instalar todas las herramientas necesarias de forma manual, como por ejemplo, mediante contenedores de Docker.

8.3. VÍAS FUTURAS

La aplicación que se ha desarrollado permite que se puedan escalar múltiples funciones, al seguir un patrón MVC con una gran modularización. Entre las posibles futuras funciones o mejoras están:

- Guardar en la BD las métricas y pesos predichos, ya que hasta ahora estos se almacenan temporalmente mediante ficheros *.txt*.
- Que la recogida de los datos de los experimentos no sólo sea mediante la lectura de un CSV, sino mediante una conexión directa con el sensor de peso.
- Reducir el tiempo de carga de los algoritmos.
- Optimizar el algoritmo usando Prophet o cambiando por completo o parcialmente la estrategia de predicción mediante el uso de otras librerías de Machine Learning.
- Desplegar la aplicación con Docker.
- Crear una versión móvil de la aplicación web.
- Implementar otros ajustes de curva distintos (con ecuaciones polinómicas de otros grados).
- Otros: mejorar el diseño de la web, generar nuevas vistas, etc.

9. BIBLIOGRAFÍA

- Acaia. (20 de Diciembre de 2020). *Acaia*. Obtenido de Pearl: <https://acaia.co/collections/coffee-scales/products/pearl?variant=2433774125079>
- Amor García, Á., Guardiola Elías, J., & Marín García, F. (13 de Marzo de 2019). *UCAM Campus Virtual*. Obtenido de Metodología Kanban: <https://cv.ucam.edu/access/content/group/18ULJ792P28/Metodolog%C3%ADas%20Primer%20Parcial/Kanban.pptx>
- Anaconda. (15 de Mayo de 2021). *Anaconda*. Obtenido de Your data science toolkit: <https://www.anaconda.com/products/individual#windows>
- Anaconda. (3 de Marzo de 2021). *Anaconda Documentation*. Obtenido de Anaconda Navigator: <https://docs.anaconda.com/anaconda/navigator/>
- Anh Khoa, T., Hoang Phuc, C., Duc Lam, P., Bao Nhu, L., Minh Trong, N., Thi Hoang Phuong, N., . . . Ngoc Minh Duc, D. (2020). Waste management system using IoT-based machine learning in university. *Wireless Communications and Mobile Computing, 2020*, 1-13.
- Apache Friends. (28 de Diciembre de 2020). *Apache Friends*. Obtenido de XAMPP Apache + MariaDB + PHP + Perl: <https://www.apachefriends.org/es/index.html>
- Apache Friends. (29 de Mayo de 2021). *Apache Friends*. Obtenido de XAMPP Apache + MariaDB + PHP + Perl: <https://www.apachefriends.org/es/index.html>
- Bancroft-Connors, J. (1 de Mayo de 2020). *Applied Frameworks*. Obtenido de Agile Metrics: 4 Balanced KPIs to Measure Success: <https://appliedframeworks.com/agile-metrics-4-balanced-kpis-to-measure-success/>
- BidDown. (28 de Diciembre de 2020). *BidDown*. Obtenido de ¿Qué es el Gasto Indirecto? – Procurement Topics: <https://biddown.com/que-es-el-gasto-indirecto-procurement-topics/#:~:text=Es%20cualquier%20gasto%20realizado%20por,dar%20continuidad%20a%20una%20empresa.>
- binti Md. Ishak, N., & bin Abdullah, S. (2020). Design of IoT garbage monitoring with weight sensing. *International Research Journal of Engineering and Technology, 7(7)*, 553-556.
- Bootstrap. (5 de Enero de 2021). *Bootstrap*. Obtenido de Build fast, responsive sites with Bootstrap: <https://getbootstrap.com/>
- Brown, E. (1 de Enero de 2018). *Python Data*. Obtenido de Forecasting Time Series data with Prophet – Part 4: <https://pythondata.com/forecasting-time-series-data-prophet-part-4/>
- Brown, T. (14 de Octubre de 2020). *Hook Agency*. Obtenido de 7 Amazing Website Color Schemes 2021 + Color Mood Board: <https://hookagency.com/website-color-schemes/>
- Brownlee, J. (9 de Junio de 2017). *Machine Learning Mastery*. Obtenido de How to Create an ARIMA Model for Time Series Forecasting in Python: <https://machinelearningmastery.com/arima-for-time-series-forecasting-with-python/>

- Dey, R. (17 de Abril de 2019). *Visual Studio Marketplace*. Obtenido de Live Server: <https://marketplace.visualstudio.com/items?itemName=ritwickdey.LiveServer>
- Dhana Satyamanikanta, S., & Madeshan, N. (2017). Smart garbage monitoring system using sensors with RFID over internet of things. *Journal of Advanced Research in Dynamical and Control Systems*, 9(6), 133-140.
- Elliott, E. (29 de Agosto de 2015). *Medium*. Obtenido de 5 Questions Every Unit Test Must Answer: <https://medium.com/javascript-scene/what-every-unit-test-needs-f6cd34d9836d#.r6jx21roc>
- Escuela de negocios EBF. (11 de Septiembre de 2019). *EBF*. Obtenido de Ventajas y desventajas de las metodologías Agile (ágiles): <https://ebf.com.es/blog/ventajas-y-desventajas-de-las-metodologias-agiles-y-su-aplicacion-en-el-trabajo/>
- Fenopix. (8 de Enero de 2021). *canvasJS*. Obtenido de Beautiful Charts for your PHP Applications: <https://canvasjs.com/php-charts/>
- Gálvez Sánchez, A., Torres Segura, A., Saura Cuadrado, I., & Pascoe-Kelly, M. (13 de Marzo de 2019). *UCAM Campus Virtual*. Obtenido de Scrum Ban: <https://cv.ucam.edu/access/content/group/18ULJ792P28/Metodolog%C3%ADas%20Pramer%20Parcial/Scrumban%202.pdf>
- glassdoor. (19 de Enero de 2021). *glassdoor*. Obtenido de Sueldos para Scrum Master: https://www.glassdoor.es/Sueldos/scrum-master-sueldo-SRCH_KOO,12.htm
- Gómez, R. (11 de Noviembre de 2015). *Rodrigogr*. Obtenido de Modelo Vista Controlador: <http://rodrigogr.com/blog/modelo-vista-controlador/#:~:text=Ventajas%20de%20MVC&text=La%20separaci%C3%B3n%20del%20Modelo%20y,m%C3%BAltiples%20representaciones%20de%20los%20datos.>
- Google. (9 de Febrero de 2021). *Te damos la bienvenida a Colaboratory*. Obtenido de ¿Qué es Colaboratory?: <https://colab.research.google.com/notebooks/welcome.ipynb?hl=es>
- Guiadev. (04 de Febrero de 2019). *Guiadev - Cursos y Tutoriales de Desarrollo Web en Español*. Obtenido de PHP vs JAVA: <https://guiadev.com/php-vs-java/>
- Guru99. (20 de Diciembre de 2020). *Guru99*. Obtenido de R Vs Python: What's the Difference?: <https://www.guru99.com/r-vs-python.html#:~:text=R%20and%20Python%20are%20both,general%20approach%20to%20data%20science.&text=Python%20is%20a%20general%2Dpurpose%20language%20with%20a%20readable%20syntax>
- Hack a Boss. (12 de Julio de 2019). *Hack a Boss*. Obtenido de Salario de programador en España 2018/2019. También 2020.: <https://hackaboss.com/blog/salario-programador-espana-2018-2019/#:~:text=Para%20personas%20que%20est%C3%A1n%20empezando,salario%20ronda%20los%2040.000%20euros.>
- Izquierdo, J., Gómez, J., & Nava, K. (13 de Marzo de 2019). *Ucam Campus Virtual*. Obtenido de Scrum Metodología Ágil: <https://cv.ucam.edu/access/content/group/18ULJ792P28/Metodolog%C3%ADas%20Pramer%20Parcial/Presentaci%C3%B3n%20SCRUM.pptx>

Aplicación web para la visualización de datos de sensor “Smart Bin”

Karagiannis, T. (21 de Septiembre de 2014). *Stack Overflow*. Obtenido de Change form action on select option: <https://stackoverflow.com/questions/25960729/change-form-action-on-select-option>

Kzi-Kaizenia. (20 de Diciembre de 2020). *Kzi-Kaizenia*. Obtenido de ¿Cómo se estima en Scrum?: <https://kzi.mx/scrum/#:~:text=La%20estimaci%C3%B3n%20de%20costos%20se,puede%20utilizar%20puntos%20de%20historia>

Matt. (18 de Octubre de 2011). *Stack Overflow*. Obtenido de How to create a simple PHP cookie language toggle?: <https://stackoverflow.com/questions/7791126/how-to-create-a-simple-php-cookie-language-toggle>

Mewburn, B. (4 de Enero de 2021). *Visual Studio Marketplace*. Obtenido de PHP Intelephense: <https://marketplace.visualstudio.com/items?itemName=bmewburn.vscode-intelephense-client>

Microsoft. (30 de Diciembre de 2020). *Visual Studio Marketplace*. Obtenido de Code editing. Redefined: <https://code.visualstudio.com/>

Microsoft. (16 de Mayo de 2021). *Microsoft*. Obtenido de Microsoft Excel: <https://www.microsoft.com/es-es/microsoft-365/excel>

Microtronics Technologies. (15 de Diciembre de 2020). *Microtronics Technologies – Projects of 8051, Arduino, Raspberry Pi*. Recuperado el 15 de Diciembre de 2020, de IOT based Garbage Monitoring Using Arduino: <https://www.projects8051.com/iot-based-garbage-monitoring-using-arduino/>

Monday.com. (16 de Mayo de 2021). *Monday.com*. Obtenido de Trabaja de la mejor manera para ti: <https://monday.com/lang/es/>

NeuronForest. (20 de Diciembre de 2020). *NeuronForest*. Obtenido de ¿Cuánto debe durar mi sprint en Scrum?: <https://neuronforest.es/cuanto-debe-durar-mi-sprint-en-scrum/#:~:text=Los%20sprints%20son%20las%20iteraciones,%2C%20y%20m%C3%A1ximo%204%20semanas>

Parrales Cáceres, J., Sepúlveda Madrid, B., Fernández Segura, S., & Parra Navarro, L. (13 de Marzo de 2019). *UCAM Campus Virtual*. Obtenido de Programación Extrema XP: <https://cv.ucam.edu/access/content/group/18ULJ792P28/Metodolog%C3%ADas%20Parral%20Parcial/Metodolog%C3%ADaXP.pptx>

phpMyAdmin. (28 de Diciembre de 2020). *phpMyAdmin*. Obtenido de Bringing MySQL to the web: <https://www.phpmyadmin.net/>

phpspot. (11 de Abril de 2020). *phpspot helping build websites*. Obtenido de Import CSV File into MySQL using PHP: <https://phpspot.com/php/import-csv-file-into-mysql-using-php/#:~:text=HTML%20Form%20to%20Import%20CSV%20File&text=After%20submitting%20the%20file%2C%20PHP,type%20is%20restricted%20by%20specifying%20>

Prakash, P. (23 de Noviembre de 2020). *Visual Studio Marketplace*. Obtenido de CSS Peek: <https://marketplace.visualstudio.com/items?itemName=pranaygp.vscode-css-peek>

- Prettier. (20 de Noviembre de 2020). *Visual Studio Marketplace*. Obtenido de Prettier - Code formatter: <https://marketplace.visualstudio.com/items?itemName=esbenp.prettier-vscode>
- Project Jupyter. (8 de Febrero de 2021). *Jupyter*. Obtenido de Jupyter: <https://jupyter.org/>
- Prophet. (19 de Febrero de 2021). *Prophet Docs*. Obtenido de Trend Changepoints: https://facebook.github.io/prophet/docs/trend_changepoints.html#adjusting-trend-flexibility
- Rex, S. (4 de Marzo de 2020). *Visual Studio Marketplace*. Obtenido de PHPDoc Comment: <https://marketplace.visualstudio.com/items?itemName=rexshi.phpdoc-comment-vscode-plugin>
- Robles, V. (5 de Enero de 2021). *Victor Robles | Blog sobre Desarrollo web y Administración de sistemas*. Obtenido de MVC (Modelo Vista Controlador) en PHP nativo: <https://victorroblesweb.es/2013/11/18/tutorial-mvc-en-php-nativo/>
- Schwaber, K., & Sutherland, J. (1 de Julio de 2013). *La Guía de Scrum*. Obtenido de La Guía Definitiva de Scrum: Las Reglas del Juego: <https://www.scrumguides.org/docs/scrumguide/v1/scrum-guide-es.pdf>
- Singh, A. (30 de Agosto de 2018). *Analytics Vidhya*. Obtenido de Build High Performance Time Series Models using Auto ARIMA in Python and R: <https://www.analyticsvidhya.com/blog/2018/08/auto-arima-time-series-modeling-python-r/>
- SonarSource. (17 de Diciembre de 2020). *Visual Studio Marketplace*. Obtenido de SonarLint: <https://marketplace.visualstudio.com/items?itemName=SonarSource.sonarlint-vscode>
- Subit, A. (3 de Abril de 2019). *Stack Overflow*. Obtenido de How to change XAMPP apache server port?: <https://stackoverflow.com/questions/11294812/how-to-change-xampp-apache-server-port>
- Taylor, S. J., & Letham, B. (2017). Forecasting at Scale. *The American Statistician*, 72(1), 37-45.
- TeamGantt. (16 de Mayo de 2021). *TeamGantt*. Obtenido de Get a top-rated gantt chart for free, forever.: <https://www.teamgantt.com/>
- Wikipedia. (2 de Enero de 2020). *Wikipedia*. Obtenido de Desarrollo guiado por pruebas: [https://es.wikipedia.org/wiki/Desarrollo_guiado_por_pruebas#:~:text=Desarrollo%20guiado%20por%20pruebas%20de%20software%2C%20o%20Test%2Ddriven%20development,%20y%20Refactorizaci%C3%B3n%20\(Refactoring\)](https://es.wikipedia.org/wiki/Desarrollo_guiado_por_pruebas#:~:text=Desarrollo%20guiado%20por%20pruebas%20de%20software%2C%20o%20Test%2Ddriven%20development,%20y%20Refactorizaci%C3%B3n%20(Refactoring)).
- Wikipedia. (6 de Mayo de 2021). *Wikipedia*. Obtenido de Curve fitting: https://en.wikipedia.org/wiki/Curve_fitting
- ZeroSpectrum. (2 de Junio de 2019). *ZeroSpectrum*. Obtenido de MAE vs MSE vs RMSE: <http://zerospectrum.com/2019/06/02/mae-vs-mse-vs-rmse/>

10. ANEXOS

10.1. MANUAL DE INSTALACIÓN

A continuación, se va a mostrar la instalación en local de las herramientas necesarias para que la web funcione en su plenitud.

10.1.1. DESCARGA E INSTALACIÓN DE XAMPP

Como se comentó en el apartado 4.1 (*Herramientas software para la aplicación web*), XAMPP va a ser necesario para albergar la página web en el servidor Apache y la base de datos con phpMyAdmin (administrador de MySQL).

En primer lugar, se debe **descargar XAMPP** para Windows (Apache Friends, 2021) pulsando el botón de descarga que aparece en la parte inferior en la ilustración 56.



Ilustración 56. Instalación: descargar XAMPP

Una vez descargado, se procede con la **instalación de XAMPP**, con lo que solamente deben seleccionar los módulos Apache, MySQL, PHP y phpMyAdmin de la ilustración 57.

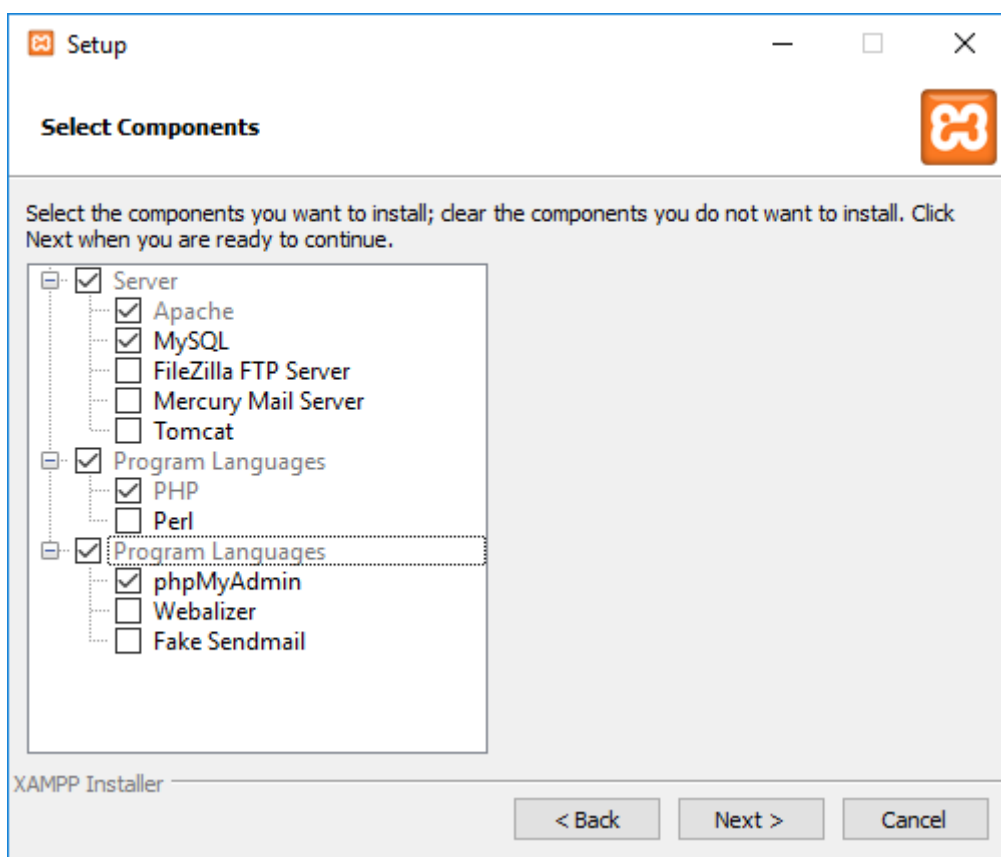


Ilustración 57. Instalación: instalar XAMPP 1

Para finalizar con la instalación, se selecciona el fichero donde se quiera instalar XAMPP como aparece en la ilustración 58.

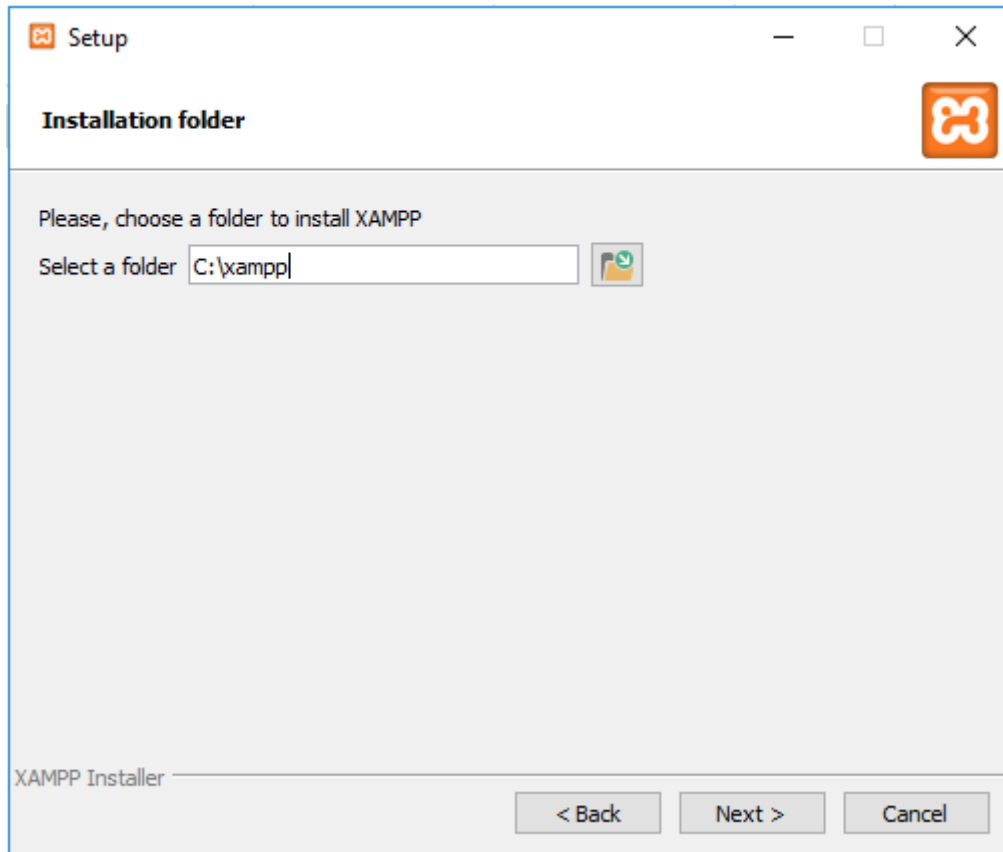


Ilustración 58. Instalación: instalar XAMPP 2

10.1.2. DESCARGA DEL CÓDIGO FUENTE Y DE LOS FICHEROS DE INSTALACIÓN

Para poder empezar a configurar Apache y phpMyAdmin, se deben **descargar el código fuente y los ficheros de instalación** en <https://github.com/AlbertoPonsButtazzo/smartbin.git>. Se debe pulsar a *Download ZIP* tal como aparece en la ilustración 59.

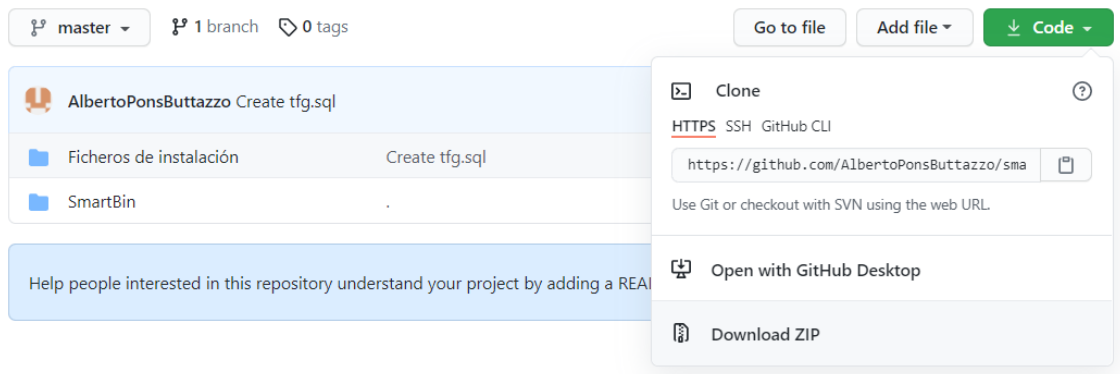


Ilustración 59. Instalación: descargar código fuente

Una vez descargado el fichero, este se debe descomprimir en el directorio donde se instaló XAMPP, más concretamente en la carpeta *htdocs* de la ilustración 60.

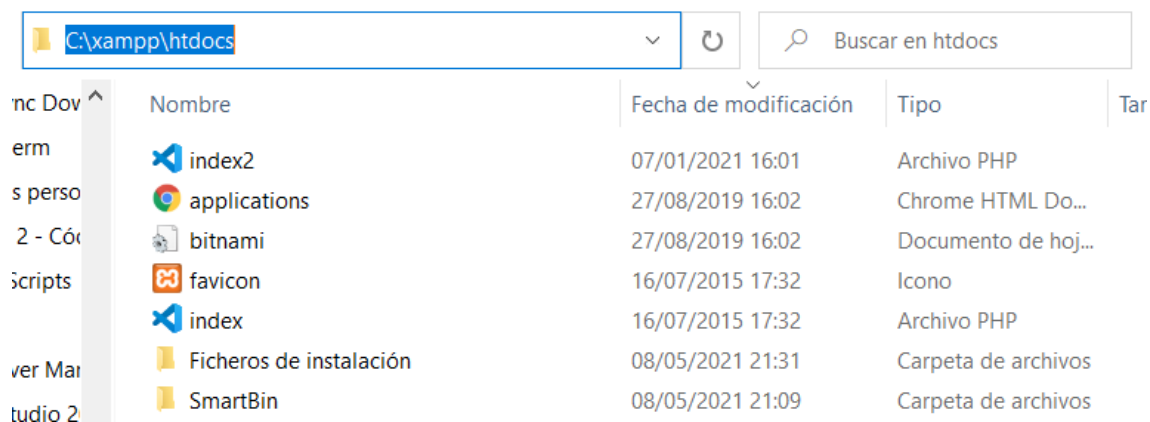


Ilustración 60. Instalación: guardar ficheros

10.1.3. CONFIGURACIÓN DE APACHE

Para **configurar Apache**, se debe abrir la carpeta *Ficheros de instalación* y sustituir el fichero *httpd* de dicha carpeta por el fichero *httpd* de la carpeta *apache/conf/* de la ilustración 61. Además, debe sustituir también el fichero *httpd-ssl* por el fichero *httpd-ssl* de la carpeta *apache/conf/extra* de la ilustración 62.

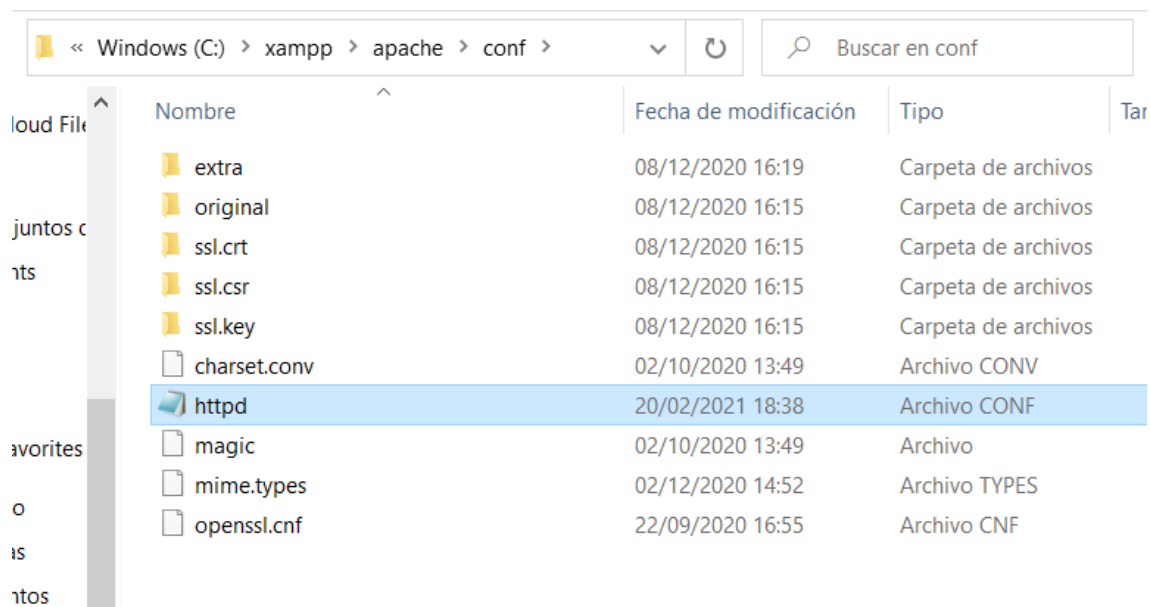


Ilustración 61. Instalación: modificar ficheros 1

Aplicación web para la visualización de datos de sensor “Smart Bin”

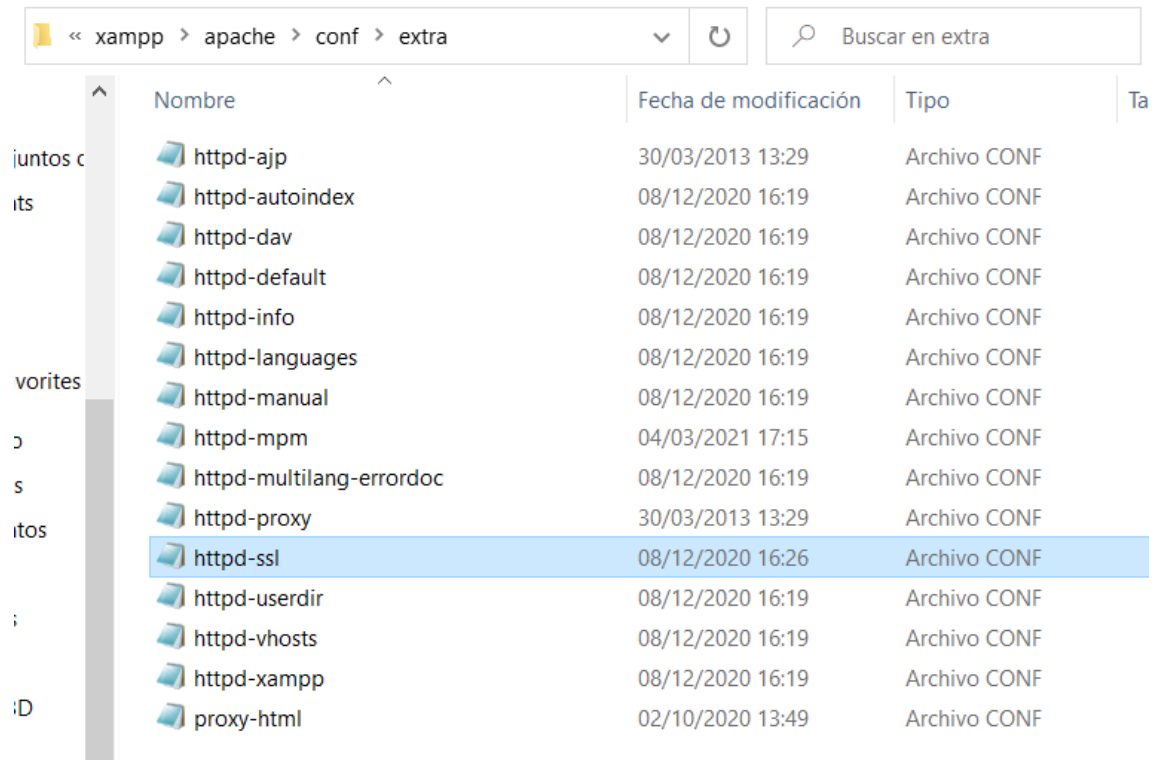


Ilustración 62. Instalación: modificar ficheros 2

Después de realizar dichos cambios, puede ejecutar el programa XAMPP e iniciar Apache y MySQL pulsando el botón *start* de ambos módulos, como en la ilustración 63.

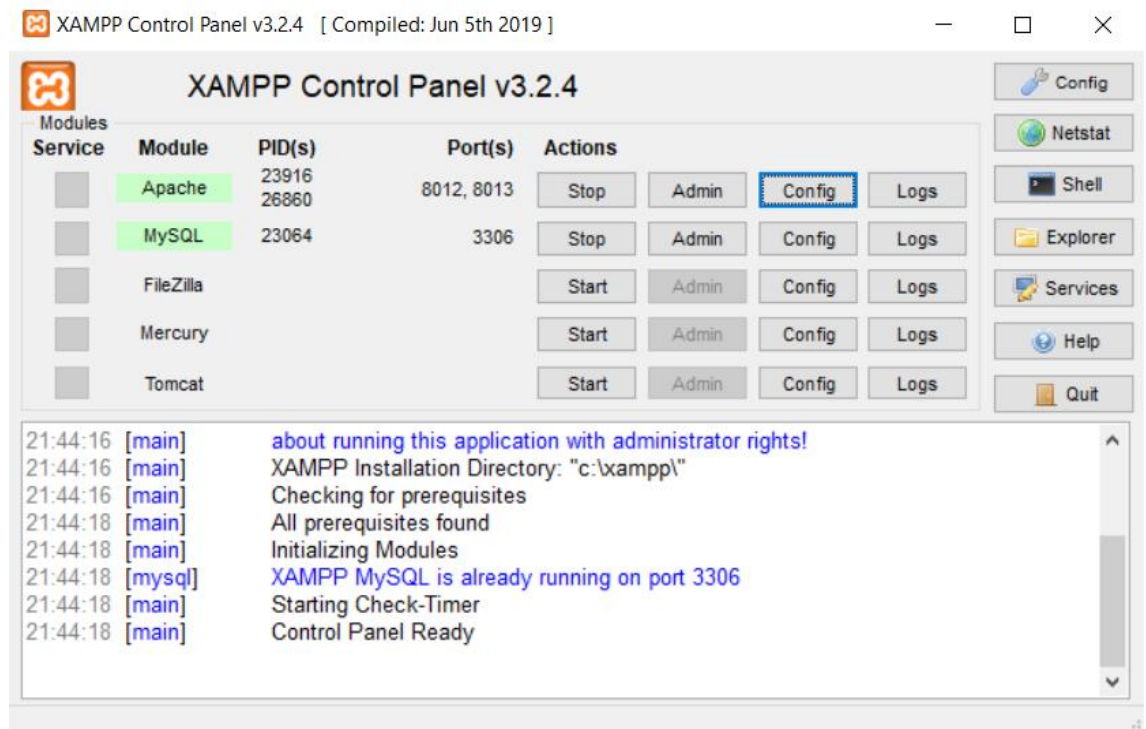


Ilustración 63. Instalación: iniciar Apache y MySQL

10.1.4. CONFIGURACIÓN DE PHPMYADMIN

Para poder **usar la base de datos**, se tiene que importar el fichero *tfg.sql* también disponible en la carpeta *Ficheros de instalación*. Para ello, debe acceder a *phpMyAdmin* escribiendo en su navegador *localhost:8012/phpmyadmin/*, como aparece en la ilustración 64. Una vez dentro, debe crear una nueva base de datos pulsado a *Nueva* en la parte superior izquierda de la web. Y se debe crear con el nombre *tfg* y con el cotejamiento *utf8_bin*, como en la ilustración 65.



Ilustración 64. Instalación: crear tabla BD 1

Bases de datos

Crear base de datos

| | Base de datos | Cotejamiento | Acción |
|--------------------------|--------------------|--------------------|-------------------------|
| <input type="checkbox"/> | information_schema | utf8_general_ci | Seleccionar privilegios |
| <input type="checkbox"/> | mysql | utf8mb4_general_ci | Seleccionar privilegios |
| <input type="checkbox"/> | performance_schema | utf8_general_ci | Seleccionar privilegios |
| <input type="checkbox"/> | phpmyadmin | utf8_bin | Seleccionar privilegios |
| <input type="checkbox"/> | sql | utf8_bin | Seleccionar privilegios |

Ilustración 65. Instalación: crear tabla BD 2

Para finalizar con la configuración de *phpMyAdmin*, se debe importar el fichero *tfg.sql*, también disponible en la carpeta *Ficheros de instalación*. Para ello, se accede al menú *Importar* en la parte superior central, se selecciona el archivo y se pulsa el botón *Continuar* de la parte inferior derecha, como en la ilustración 66.

Aplicación web para la visualización de datos de sensor “Smart Bin”

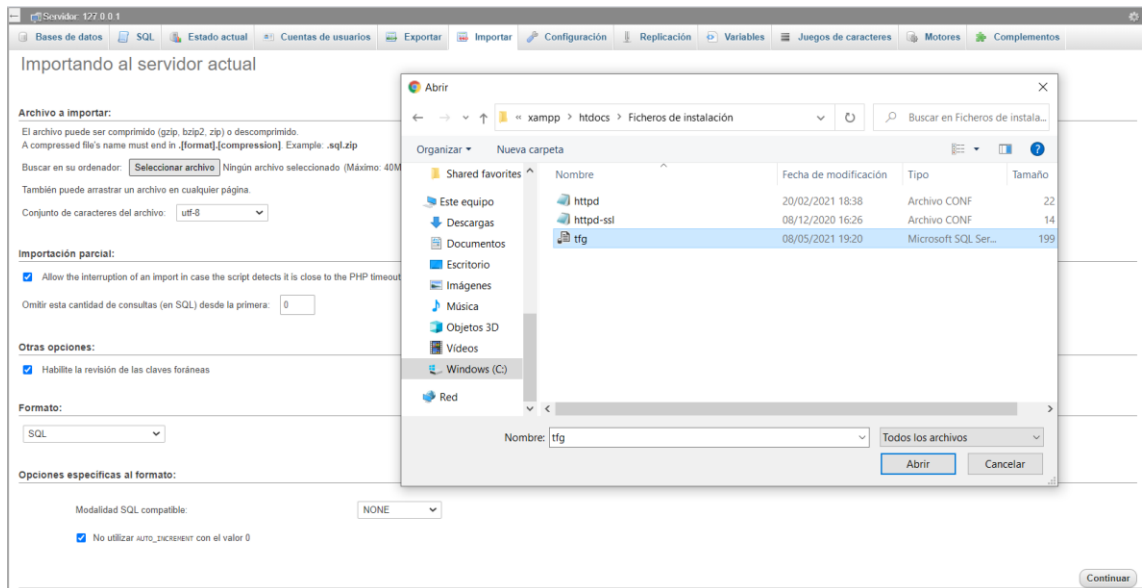


Ilustración 66. Instalación: importar BD

10.1.5. DESPLIEGUE DE LA WEB

Cuando se completan los pasos anteriores, ya se puede **acceder a la web sin poder usar la aplicación inteligente**. Para ello, se debe escribir en el navegador *localhost:8012/SmartBin* como en la ilustración 67.



Ilustración 67. Instalación: despliegue de la web

Para poder usar la aplicación inteligente se deben seguir los pasos que aparecen a continuación.

10.1.6. CONFIGURACIÓN DEL ENTORNO ANACONDA

Si se desea usar la aplicación inteligente, se debe **configurar el entorno de Anaconda** primero. Para ello, se debe descargar e instalar la aplicación Anaconda Navigator en la página oficial de Anaconda (Anaconda, 2021). En primer lugar, descargue el instalador 64-Bit Graphical Installer que aparece al final de la web, aquel que aparece a la izquierda de la ilustración 68.



Ilustración 68. Instalación: descargar Anaconda Navigator

En la instalación, seleccione la versión individual y elija un directorio de almacenamiento, como en las ilustraciones 69 y 70.

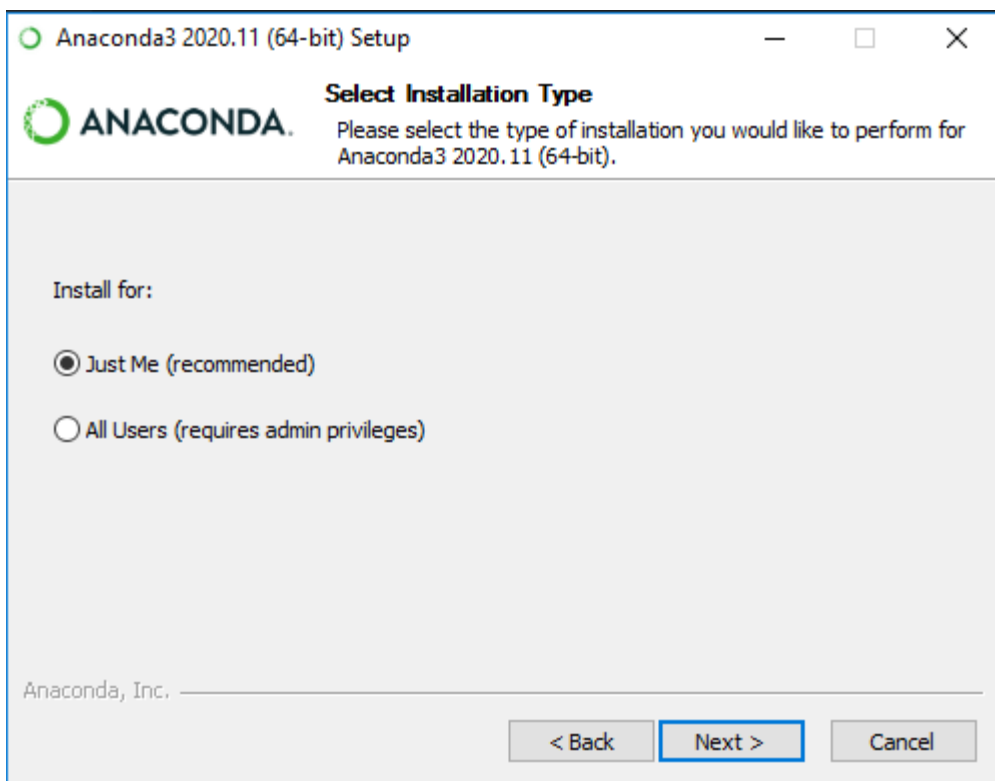


Ilustración 69. Instalación: instalar Anaconda Navigator 1

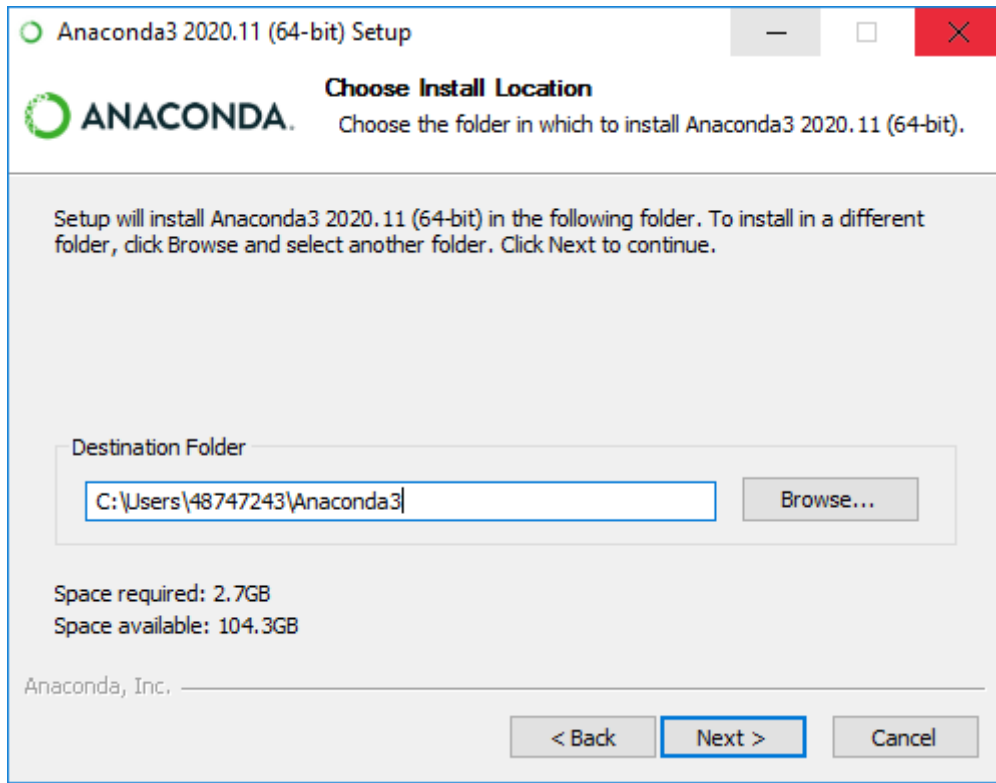


Ilustración 70. Instalación: instalar Anaconda Navigator 2

Tras realizar la descarga, se debe modificar el entorno por defecto para que se pueda ejecutar la aplicación inteligente, siendo necesario importar las librerías:

- *python* (3.8.5): necesario para ejecutar la aplicación inteligente.
- *scikit-learn* (0.24.1): necesario para realizar los cálculos matemáticos para generar las métricas.
- *pandas* (1.2.3): necesario para generar un *dataframe* (estructura de datos de dos dimensiones).
- *numpy* (1.19.2): también necesario para realizar cálculos matemáticos.
- *pystan* (2.19.1.1): necesario para realizar modelado y computación estadística con Python.
- *fbprophet* (0.7.1): necesario para usar Prophet.

Para poder importar dichas librerías, es necesario abrir Anaconda Navigator y modificar el entorno *root*. En la página de inicio, seleccione la opción

root en la parte superior y ejecute la primera aplicación *CMD.exe Prompt*, como se muestra en la ilustración 71.

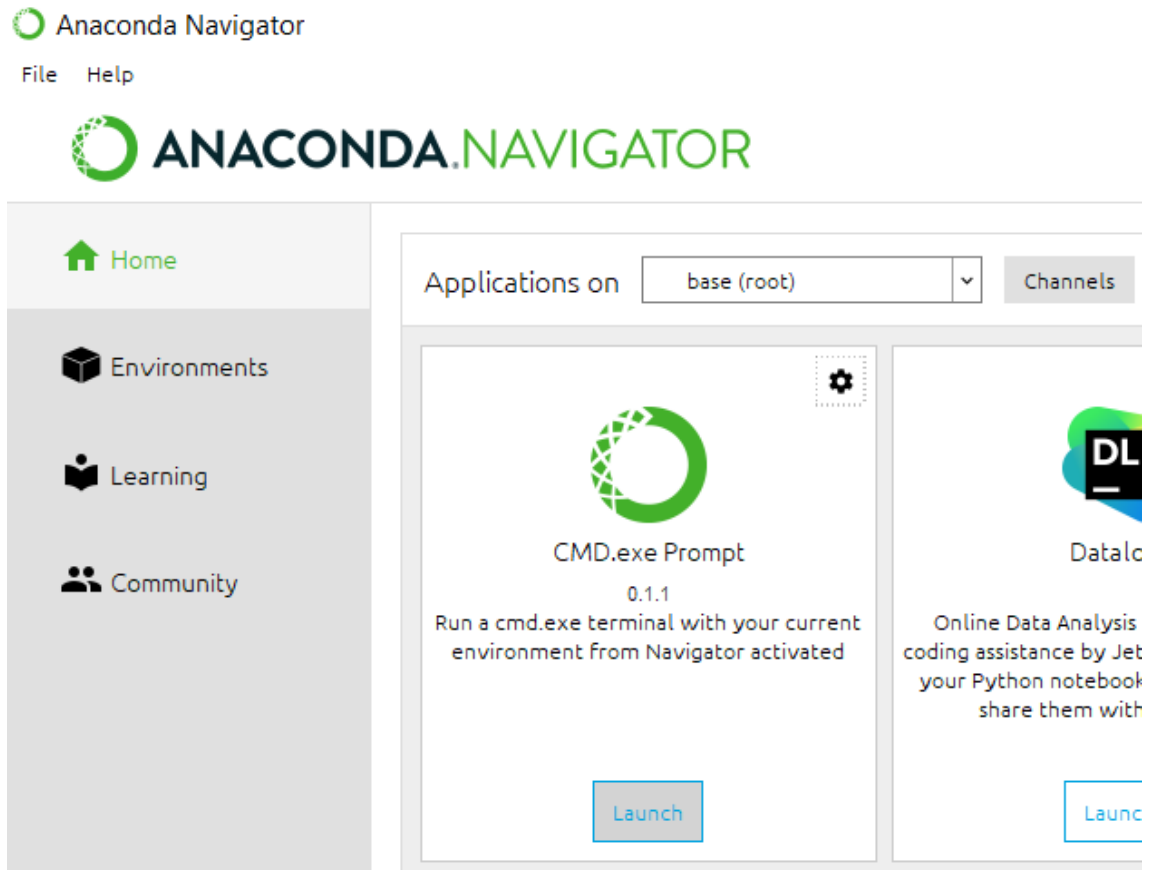


Ilustración 71. Instalación: ejecutar entorno root

Tras realizar estas acciones, aparece un terminal donde se deben escribir las siguientes directivas, para poder instalar las librerías y hacer funcionar la aplicación de forma local:

- *conda init powershell* (ilustración 72): permite que el *PowerShell* pueda ejecutar la aplicación inteligente con Anaconda.

Aplicación web para la visualización de datos de sensor “Smart Bin”

```
C:\Windows\system32\cmd.exe
Microsoft Windows [Versión 10.0.19042.867]
(c) 2020 Microsoft Corporation. Todos los derechos reservados.

(prueba) C:\Users\alber>conda init powershell
no change C:\Users\alber\anaconda3\Scripts\conda.exe
no change C:\Users\alber\anaconda3\Scripts\conda-env.exe
no change C:\Users\alber\anaconda3\Scripts\conda-script.py
no change C:\Users\alber\anaconda3\Scripts\conda-env-script.py
no change C:\Users\alber\anaconda3\condabin\conda.bat
no change C:\Users\alber\anaconda3\Library\bin\conda.bat
no change C:\Users\alber\anaconda3\condabin\conda_activate.bat
no change C:\Users\alber\anaconda3\condabin\rename_tmp.bat
no change C:\Users\alber\anaconda3\condabin\conda_auto_activate.bat
no change C:\Users\alber\anaconda3\condabin\conda_hook.bat
no change C:\Users\alber\anaconda3\Scripts\activate.bat
no change C:\Users\alber\anaconda3\condabin\activate.bat
no change C:\Users\alber\anaconda3\condabin\deactivate.bat
no change C:\Users\alber\anaconda3\Scripts\activate
no change C:\Users\alber\anaconda3\Scripts\deactivate
no change C:\Users\alber\anaconda3\etc\profile.d\conda.sh
no change C:\Users\alber\anaconda3\etc\fish\conf.d\conda.fish
no change C:\Users\alber\anaconda3\shell\condabin\Conda.psm1
no change C:\Users\alber\anaconda3\shell\condabin\conda-hook.ps1
no change C:\Users\alber\anaconda3\Lib\site-packages\xontrib\conda.xsh
no change C:\Users\alber\anaconda3\etc\profile.d\conda.csh
no change C:\Users\alber\Documents\WindowsPowerShell\profile.ps1
No action taken.

(prueba) C:\Users\alber>
```

Ilustración 72. Instalación: comando de modificación del entorno 1

- `conda install -c conda-forge fbprophet` (ilustraciones 73 y 74):
librería de Prophet.

```
C:\Windows\system32\cmd.exe - conda install -c conda-forge fbprophet
(prueba) C:\Users\alber>conda install -c conda-forge fbprophet
Collecting package metadata (current_repodata.json): done
Solving environment: done

==> WARNING: A newer version of conda exists. <==
  current version: 4.9.2
  latest version: 4.10.1

Please update conda by running

  $ conda update -n base -c defaults conda

## Package Plan ##

environment location: C:\Users\alber\anaconda3\envs\prueba

added / updated specs:
- fbprophet

The following packages will be downloaded:

package | build | size | channel
-----|-----|-----|-----
arviz-0.11.2 | pyhd8ed1ab_1 | 1.4 MB | conda-forge
bzip2-1.0.8 | h8ffe710_4 | 149 KB | conda-forge
convertdate-2.3.2 | pyhd8ed1ab_0 | 37 KB | conda-forge
```

Ilustración 73. Instalación: comando de modificación del entorno 2.1

```

C:\Windows\system32\cmd.exe - conda install -c conda-forge fbprophet
Proceed ([y]/n)? y

Downloading and Extracting Packages
m2w64-gcc-libgfortra | 342 KB | ##### | 100%
tqdm-4.60.0 | 79 KB | ##### | 100%
m2w64-gmp-6.1.0 | 726 KB | ##### | 100%
packaging-20.9 | 35 KB | ##### | 100%
xz-5.2.5 | 211 KB | ##### | 100%
pandas-1.2.4 | 10.2 MB | ##### | 100%
zlib-1.2.11 | 126 KB | ##### | 100%
libblas-3.9.0 | 4.0 MB | ##### | 100%
lz4-c-1.9.3 | 134 KB | ##### | 100%
arviz-0.11.2 | 1.4 MB | ##### | 100%
bzip2-1.0.8 | 149 KB | ##### | 100%
m2w64-gcc-libs-5.3.0 | 520 KB | ##### | 100%
libzip-1.7.3 | 134 KB | ##### | 100%
lcms2-2.12 | 882 KB | ##### | 100%
numpy-1.20.2 | 5.3 MB | ##### | 100%
holidays-0.11.1 | 68 KB | ##### | 100%
krb5-1.19.1 | 839 KB | ##### | 100%
olefile-0.46 | 32 KB | ##### | 100%
netcdf4-1.5.6 | 383 KB | ##### | 100%
curl-7.76.1 | 128 KB | ##### | 100%
libnetcdf-4.8.0 | 671 KB | ##### | 100%
openjpeg-2.4.0 | 238 KB | ##### | 100%
cython-0.29.23 | 1.9 MB | ##### | 100%
liblapack-3.9.0 | 4.0 MB | ##### | 100%
tornado-6.1 | 649 KB | ##### | 100%
pyparsing-2.4.7 | 60 KB | ##### | 100%

```

Ilustración 74. Instalación: comando de modificación del entorno 2.2

- *conda install -c anaconda numpy=1.19.2* (ilustraciones 75 y 76):
librería *Numpy*.

```

C:\Windows\system32\cmd.exe - conda install -c conda-forge fbprophet - conda install -c anaconda numpy - conda install -c anac...
(prueba) C:\Users\alber>conda install -c anaconda numpy=1.19.2
Collecting package metadata (current_repodata.json): done
Solving environment: failed with initial frozen solve. Retrying with flexible solve.
Collecting package metadata (repodata.json): done
Solving environment: done

==> WARNING: A newer version of conda exists. <==
  current version: 4.9.2
  latest version: 4.10.1

Please update conda by running

  $ conda update -n base -c defaults conda

## Package Plan ##

environment location: C:\Users\alber\anaconda3\envs\prueba

added / updated specs:
- numpy=1.19.2

The following packages will be downloaded:

package | build
-----|-----
blas-1.0 | mkl 6 KB anaconda

```

Ilustración 75. Instalación: comando de modificación del entorno 3.1

Aplicación web para la visualización de datos de sensor “Smart Bin”

```
C:\Windows\system32\cmd.exe - conda install -c conda-forge fbprophet - conda install -c anaconda numpy - conda install -c anac...
Proceed ([y]/n)? y

Downloading and Extracting Packages
mkl-2019.4 | 157.5 MB | ##### | 100%
icc_rt-2019.0.0 | 9.4 MB | ##### | 100%
scipy-1.5.0 | 15.3 MB | ##### | 100%
blas-1.0 | 6 KB | ##### | 100%
mkl-service-2.3.0 | 59 KB | ##### | 100%
mkl_random-1.1.0 | 285 KB | ##### | 100%
mkl_fft-1.2.0 | 139 KB | ##### | 100%
Preparing transaction: done
Verifying transaction: done
Executing transaction: done

(prueba) C:\Users\alber>
```

Ilustración 76. Instalación: comando de modificación del entorno 3.2

- `conda install pandas=1.2.3` (ilustraciones 77 y 78): librería *Pandas*.

```
C:\Windows\system32\cmd.exe - conda install -c conda-forge fbprophet - conda install -c anaconda numpy - conda install -c anac...
(prueba) C:\Users\alber>conda install pandas=1.2.3
Collecting package metadata (current_repodata.json): done
Solving environment: failed with initial frozen solve. Retrying with flexible solve.
Collecting package metadata (repodata.json): done
Solving environment: done

==> WARNING: A newer version of conda exists. <==
  current version: 4.9.2
  latest version: 4.10.1

Please update conda by running

  $ conda update -n base -c defaults conda

## Package Plan ##

environment location: C:\Users\alber\anaconda3\envs\prueba

added / updated specs:
- pandas=1.2.3

The following packages will be UPDATED:

ca-certificates  anaconda::ca-certificates-2020.10.14-0 --> pkgs/main::ca-certificates-2021.4.13-haa95532_1
certifi          anaconda::certifi-2020.6.20-py38_0 --> pkgs/main::certifi-2020.12.5-py38haa95532_0
```

Ilustración 77. Instalación: comando de modificación del entorno 4.1

```
C:\Windows\system32\cmd.exe - conda install -c conda-forge fbprophet - conda install -c anaconda numpy - conda install -c anac...
Proceed ([y]/n)? y
Preparing transaction: done
Verifying transaction: done
Executing transaction: done
(prueba) C:\Users\alber>
```

Ilustración 78. Instalación: comando de modificación del entorno 4.2

- `conda install -c conda-forge scikit-learn=0.24.1` (ilustraciones 79 y 80): librería *Scikit-learn*.

```
(prueba) C:\Users\alber>conda install -c conda-forge scikit-learn=0.24.1
Collecting package metadata (current_repodata.json): done
Solving environment: done

==> WARNING: A newer version of conda exists. <==
  current version: 4.9.2
  latest version: 4.10.1

Please update conda by running

  $ conda update -n base -c defaults conda

## Package Plan ##

environment location: C:\Users\alber\anaconda3\envs\prueba

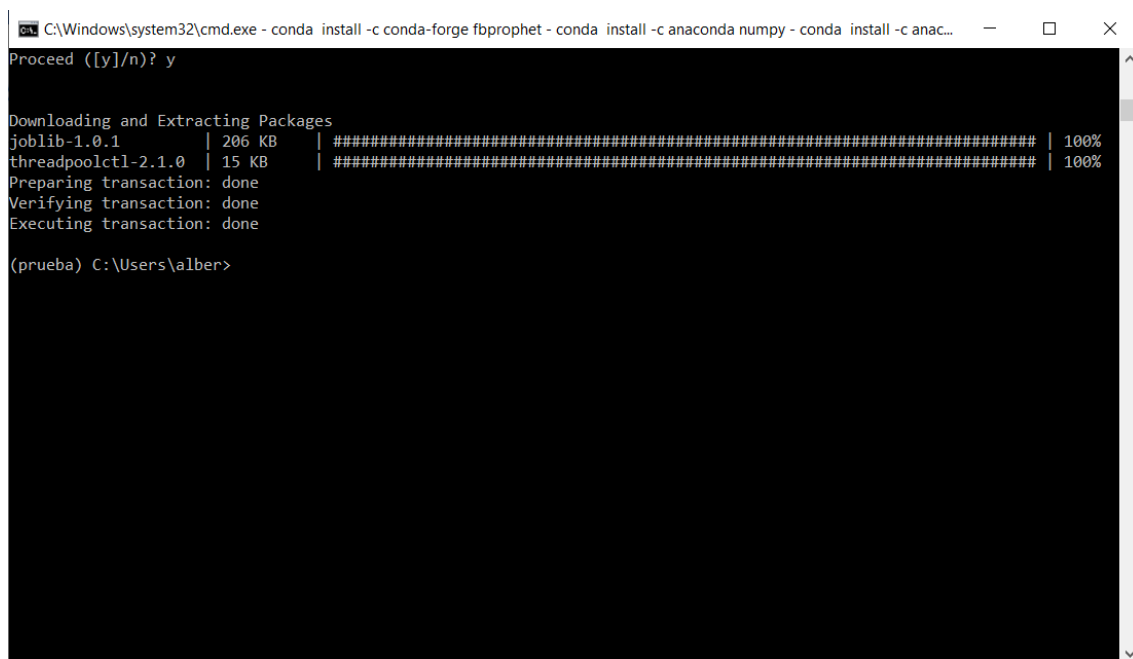
added / updated specs:
- scikit-learn=0.24.1

The following packages will be downloaded:

package | build
-----|-----
joblib-1.0.1 | pyhd8ed1ab_0 206 KB conda-forge
threadpoolctl-2.1.0 | pyh5ca1d4c_0 15 KB conda-forge
```

Ilustración 79. Instalación: comando de modificación del entorno 5.1

Aplicación web para la visualización de datos de sensor “Smart Bin”



```
C:\Windows\system32\cmd.exe - conda install -c conda-forge fbprophet - conda install -c anaconda numpy - conda install -c anac...
Proceed ([y]/n)? y

Downloading and Extracting Packages
joblib-1.0.1 | 206 KB | ##### | 100%
threadpoolctl-2.1.0 | 15 KB | ##### | 100%
Preparing transaction: done
Verifying transaction: done
Executing transaction: done

(prueba) C:\Users\alber>
```

Ilustración 80. Instalación: comando de modificación del entorno 5.2

- La librería *Pystan* ya se incluye con las anteriores librerías, así que no es necesario instalarla.

Por último, únicamente falta indicar a la aplicación inteligente el directorio donde se encuentra el *PowerShell* del sistema y el entorno de *Anaconda* que se acaba de configurar. Para ello, se debe abrir el fichero *scriptInstallation.php* del directorio donde se descargó XAMMP, más concretamente en *htdocs/SmartBin/config*, que aparece en la ilustración 81.

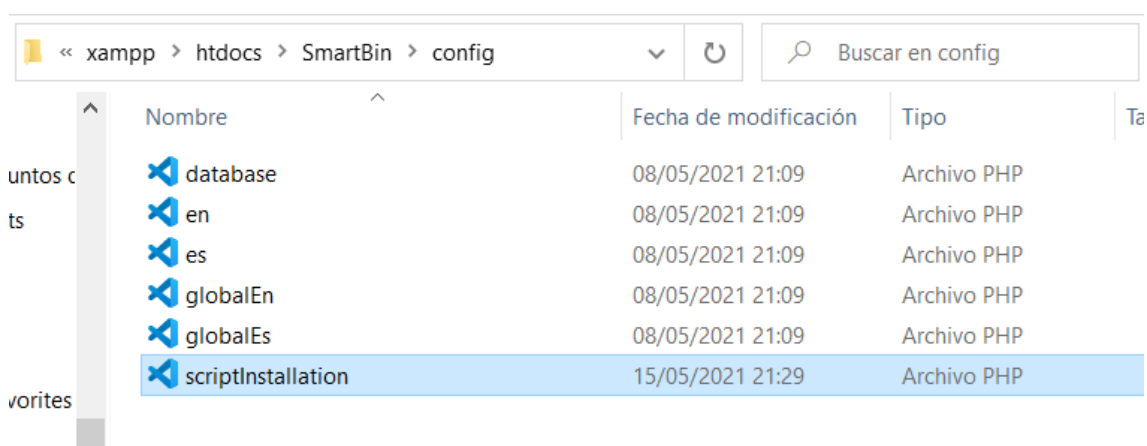
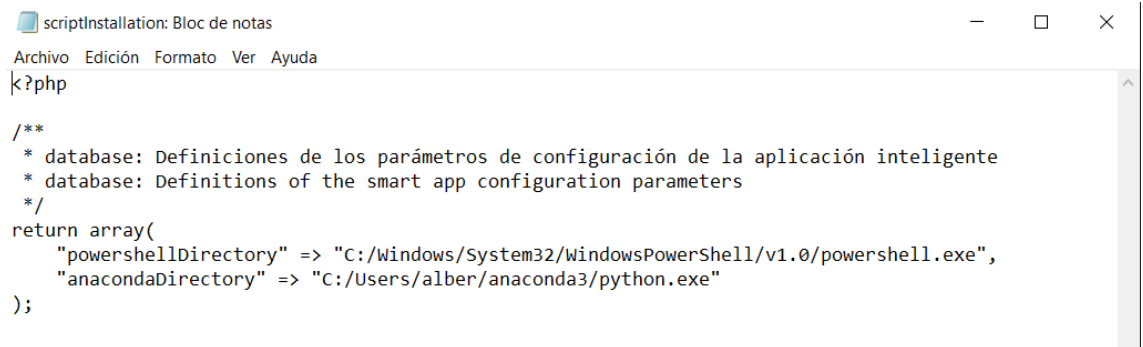


Ilustración 81. Instalación: modificar directorios 1

Se deben modificar los dos directorios en base al directorio donde está instalado Anaconda (la ruta absoluta seguido de *anaconda3/python.exe*) y

PowerShell (*C:/Windows/System32/WindowsPowerShell/v1.0/powershell.exe*), que este último por lo general será el mismo que en la ilustración 82.



```
scriptInstallation: Bloc de notas
Archivo Edición Formato Ver Ayuda
<?php

/**
 * database: Definiciones de los parámetros de configuración de la aplicación inteligente
 * database: Definitions of the smart app configuration parameters
 */
return array(
    "powershellDirectory" => "C:/Windows/System32/WindowsPowerShell/v1.0/powershell.exe",
    "anacondaDirectory" => "C:/Users/alber/anaconda3/python.exe"
);
```

Ilustración 82. Instalación: modificar directorios 2

10.2. MANUAL DE USUARIO

A continuación, se va a mostrar el funcionamiento de la aplicación web.

10.2.1. VENTANAS DE LA APLICACIÓN

Se pueden observar en la parte superior un menú de navegación que incluye 3 opciones (ilustración 83):

- Inicio: ventana principal donde se encuentra (1) la opción de importar el CSV, (2) la tabla de registros, (3) los gráficos de los pesos de un registro individual y (4) un gráfico con los pesos de múltiples registros.
- Vistas: ventana con (1) una serie de datos generales, (2) las gráficas de los pesos medios y (3) las gráficas de los pesos máximos.
- Configuración: ventana para configurar el aspecto y las opciones de los gráficos.

Aplicación web para la visualización de datos de sensor "Smart Bin"



Ilustración 83. Uso: menú de navegación

Por otro lado, en la parte superior derecha se encuentran dos botones que al pulsarlos, modifican el idioma de la web al castellano o al inglés (ilustración 84).



Ilustración 84. Uso: cambiar de idioma

10.2.2. VENTANA DE INICIO

En primer lugar, en la ventana de *Inicio* se encuentra la opción para **importar un CSV** del sensor de peso. Se debe seleccionar un archivo pulsando en *Seleccionar archivo* y posteriormente se pulsa el botón *Importar* para almacenar los registros del CSV (ilustración 85).

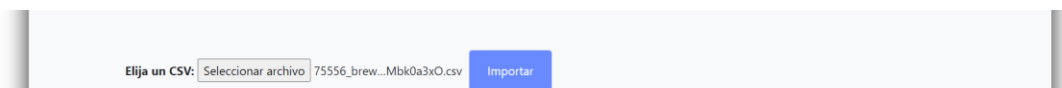


Ilustración 85. Uso: importar CSV

En segundo lugar, se encuentra **la tabla con los registros** que se han importado. De la segunda a la séptima columna se muestran los datos de cada registro (ilustración 86).

| Registros | | | | | | | | |
|--------------------------|------------|-------------|---------------------|--------------|------------|-------------|-----------------------------|--------------------------|
| Seleccionar | ID | Nombre | Creado en | Tiempo total | Peso medio | Peso máximo | Ver gráfica | Borrar |
| <input type="checkbox"/> | kSqTllqCT4 | tuesday1 | 2020-07-21 16:50:32 | 00:23:18 | 42.9 | 643.5 | Ver gráfica | <input type="checkbox"/> |
| <input type="checkbox"/> | fAFmRC5wb7 | test | 2020-09-22 14:54:59 | 00:11:14 | 5.9 | 53.099 | Ver gráfica | <input type="checkbox"/> |
| <input type="checkbox"/> | eHMc1is7KT | test | 2019-10-16 21:31:22 | 00:00:34 | 55.8 | 78.12 | Ver gráfica | <input type="checkbox"/> |
| (Sin pesos) | a9PSHhDKbr | tuesday | 2020-06-09 18:31:51 | 01:16:30 | 1073 | 1073 | (Sin pesos) | <input type="checkbox"/> |
| <input type="checkbox"/> | UDaUHwzMye | test | 2019-11-06 16:17:44 | 00:26:38 | 8.55 | 1.71 | Ver gráfica | <input type="checkbox"/> |
| <input type="checkbox"/> | S3cQcQyeJc | parentsday1 | 2020-09-26 20:31:05 | 00:15:42 | 2.67 | 24 | Ver gráfica | <input type="checkbox"/> |
| (Sin pesos) | QONLyVtLAC | hello | 2020-02-24 16:16:50 | 01:12:39 | 27.66 | 19.36 | (Sin pesos) | <input type="checkbox"/> |
| <input type="checkbox"/> | JmJ2por3Xk | parentsday3 | 2020-09-26 21:09:57 | 00:05:16 | 26 | 156 | Ver gráfica | <input type="checkbox"/> |
| (Sin pesos) | i37RdnKrTS | test | 2019-11-06 16:58:52 | 01:04:24 | 32.8 | 19.68 | (Sin pesos) | <input type="checkbox"/> |
| <input type="checkbox"/> | GihlPyhIFh | hello2 | 2020-02-24 18:58:56 | 00:16:06 | 7.14 | 35.7 | Ver gráfica | <input type="checkbox"/> |
| (Sin pesos) | ENw2qHLsJG | test2 | 2020-09-22 15:15:50 | 00:38:54 | 27.7 | 24.93 | (Sin pesos) | <input type="checkbox"/> |
| <input type="checkbox"/> | AHwGDvXm1Y | parentsday2 | 2020-09-26 20:57:10 | 00:11:56 | 3.53 | 24.7 | Ver gráfica | <input type="checkbox"/> |

Tabla con todos los registros

[Ver gráfico](#)

Ilustración 86. Uso: tabla con los registros

Si se pulsa sobre uno de los nombres de una de esas columnas, se **reorganiza la tabla de forma ascendente** en base a ese campo. En el siguiente ejemplo se organiza la tabla en base al tiempo total que dura un registro (ilustración 87).

| Registros | | | | | | | | |
|--------------------------|------------|-------------|---------------------|--------------|------------|-------------|-----------------------------|--------------------------|
| Seleccionar | ID | Nombre | Creado en | Tiempo total | Peso medio | Peso máximo | Ver gráfica | Borrar |
| <input type="checkbox"/> | eHMc1is7KT | test | 2019-10-16 21:31:22 | 00:00:34 | 55.8 | 78.12 | Ver gráfica | <input type="checkbox"/> |
| <input type="checkbox"/> | JmJ2por3Xk | parentsday3 | 2020-09-26 21:09:57 | 00:05:16 | 26 | 156 | Ver gráfica | <input type="checkbox"/> |
| <input type="checkbox"/> | fAFmRC5wb7 | test | 2020-09-22 14:54:59 | 00:11:14 | 5.9 | 53.099 | Ver gráfica | <input type="checkbox"/> |
| <input type="checkbox"/> | AHwGDvXm1Y | parentsday2 | 2020-09-26 20:57:10 | 00:11:56 | 3.53 | 24.7 | Ver gráfica | <input type="checkbox"/> |
| <input type="checkbox"/> | S3cQcQyeJc | parentsday1 | 2020-09-26 20:31:05 | 00:15:42 | 2.67 | 24 | Ver gráfica | <input type="checkbox"/> |
| <input type="checkbox"/> | GihlPyhIFh | hello2 | 2020-02-24 18:58:56 | 00:16:06 | 7.14 | 35.7 | Ver gráfica | <input type="checkbox"/> |
| <input type="checkbox"/> | kSqTllqCT4 | tuesday1 | 2020-07-21 16:50:32 | 00:23:18 | 42.9 | 643.5 | Ver gráfica | <input type="checkbox"/> |
| (Sin pesos) | pDK6FRRbo | test3 | 2020-09-26 16:48:19 | 00:26:21 | 46.4 | 27.839 | (Sin pesos) | <input type="checkbox"/> |
| <input type="checkbox"/> | UDaUHwzMye | test | 2019-11-06 16:17:44 | 00:26:38 | 8.55 | 1.71 | Ver gráfica | <input type="checkbox"/> |
| (Sin pesos) | ENw2qHLsJG | test2 | 2020-09-22 15:15:50 | 00:38:54 | 27.7 | 24.93 | (Sin pesos) | <input type="checkbox"/> |
| (Sin pesos) | i37RdnKrTS | test | 2019-11-06 16:58:52 | 01:04:24 | 32.8 | 19.68 | (Sin pesos) | <input type="checkbox"/> |
| (Sin pesos) | QONLyVtLAC | hello | 2020-02-24 16:16:50 | 01:12:39 | 27.66 | 19.36 | (Sin pesos) | <input type="checkbox"/> |
| (Sin pesos) | a9PSHhDKbr | tuesday | 2020-06-09 18:31:51 | 01:16:30 | 1073 | 1073 | (Sin pesos) | <input type="checkbox"/> |

Tabla con todos los registros

[Ver gráfico](#)

Ilustración 87. Uso: ordenar registros ascendentemente

En la parte superior de la tabla hay un buscador, que permite **buscar un registro** en base a su identificador o su nombre. En el siguiente ejemplo se busca por la palabra *parent* (ilustración 88).

Aplicación web para la visualización de datos de sensor “Smart Bin”

Se han encontrado registros con la búsqueda proporcionada.

Elija un CSV: Ningún archivo seleccionado

Registros

| Seleccionar | ID | Nombre | Creado en | Tiempo total | Peso medio | Peso máximo | Ver gráfica | Borrar |
|--------------------------|------------|-------------|---------------------|--------------|------------|-------------|-----------------------------|---------------------------------------|
| <input type="checkbox"/> | 53cQcqeJc | parentsday1 | 2020-09-26 20:31:05 | 00:15:42 | 2.67 | 24 | Ver gráfica | <input type="button" value="Borrar"/> |
| <input type="checkbox"/> | JmJ2por3Xk | parentsday3 | 2020-09-26 21:09:57 | 00:05:16 | 26 | 156 | Ver gráfica | <input type="button" value="Borrar"/> |
| <input type="checkbox"/> | AHwGDvXm1Y | parentsday2 | 2020-09-26 20:57:10 | 00:11:56 | 3.53 | 24.7 | Ver gráfica | <input type="button" value="Borrar"/> |

Tabla con todos los registros

Ilustración 88. Uso: buscar registros 1

En otro ejemplo, no se encuentran coincidencias, por lo tanto se devuelve la tabla con la totalidad de los registros (ilustración 89).

No se han encontrado resultados, se muestran todos los registros.

Elija un CSV: Ningún archivo seleccionado

Registros

| Seleccionar | ID | Nombre | Creado en | Tiempo total | Peso medio | Peso máximo | Ver gráfica | Borrar |
|--------------------------|------------|-------------|---------------------|--------------|------------|-------------|-----------------------------|---------------------------------------|
| (Sin pesos) | pDK6FfRbo | test3 | 2020-09-26 16:48:18 | 00:26:21 | 46.4 | 27.839 | (Sin pesos) | <input type="button" value="Borrar"/> |
| <input type="checkbox"/> | kSqTliqCT4 | tuesday1 | 2020-07-21 16:50:32 | 00:23:18 | 42.9 | 643.5 | Ver gráfica | <input type="button" value="Borrar"/> |
| <input type="checkbox"/> | fAFmRC5wb7 | test | 2020-09-22 14:54:59 | 00:11:14 | 5.9 | 53.099 | Ver gráfica | <input type="button" value="Borrar"/> |
| <input type="checkbox"/> | eHMctls7KT | test | 2019-10-16 21:31:22 | 00:00:34 | 55.8 | 78.12 | Ver gráfica | <input type="button" value="Borrar"/> |
| (Sin pesos) | a9PShDKbr | tuesday | 2020-06-09 18:31:51 | 01:16:30 | 1073 | 1073 | (Sin pesos) | <input type="button" value="Borrar"/> |
| <input type="checkbox"/> | UDaUHwzMye | test | 2019-11-06 16:17:44 | 00:26:38 | 8.55 | 1.71 | Ver gráfica | <input type="button" value="Borrar"/> |
| <input type="checkbox"/> | 53cQcqeJc | parentsday1 | 2020-09-26 20:31:05 | 00:15:42 | 2.67 | 24 | Ver gráfica | <input type="button" value="Borrar"/> |
| (Sin pesos) | QOnLyVLAC | hello | 2020-02-24 16:16:50 | 01:12:39 | 27.66 | 19.36 | (Sin pesos) | <input type="button" value="Borrar"/> |
| <input type="checkbox"/> | JmJ2por3Xk | parentsday3 | 2020-09-26 21:09:57 | 00:05:16 | 26 | 156 | Ver gráfica | <input type="button" value="Borrar"/> |
| (Sin pesos) | I37RdnKrTS | test | 2019-11-06 16:58:52 | 01:04:24 | 32.8 | 19.68 | (Sin pesos) | <input type="button" value="Borrar"/> |
| <input type="checkbox"/> | GihlPyhFh | hello2 | 2020-02-24 18:58:56 | 00:16:06 | 7.14 | 35.7 | Ver gráfica | <input type="button" value="Borrar"/> |
| (Sin pesos) | ENw2qHLsJG | test2 | 2020-09-22 15:15:50 | 00:38:54 | 27.7 | 24.93 | (Sin pesos) | <input type="button" value="Borrar"/> |
| <input type="checkbox"/> | AHwGDvXm1Y | parentsday2 | 2020-09-26 20:57:10 | 00:11:56 | 3.53 | 24.7 | Ver gráfica | <input type="button" value="Borrar"/> |

Tabla con todos los registros

Ilustración 89. Uso: buscar registros 2

La primera columna *Seleccionar* es para **visualizar pesos de varios registros** en un mismo gráfico. Se deben seleccionar varias casillas y posteriormente pulsar en el botón *Ver gráfico*. Se puede observar también que algunos registros no son seleccionables, esto se debe a que estos no ofrecen un registro de sus pesos (ilustración 90).

| Registros | | | | | | | | | |
|-------------------------------------|------------|-------------|---------------------|--------------|------------|-------------|-------------|--------------------------|--|
| Seleccionar | ID | Nombre | Creado en | Tiempo total | Peso medio | Peso máximo | Ver gráfica | Borrar | |
| (Sin pesos) | pDK6FtrRbo | test3 | 2020-09-26 16:48:18 | 00:26:21 | 46.4 | 27.839 | (Sin pesos) | <input type="checkbox"/> | |
| <input checked="" type="checkbox"/> | kSqTllqCT4 | tuesday1 | 2020-07-21 16:50:32 | 00:23:18 | 42.9 | 643.5 | Ver gráfica | <input type="checkbox"/> | |
| <input type="checkbox"/> | fAFmRC5wb7 | test | 2020-09-22 14:54:59 | 00:11:14 | 5.9 | 53.099 | Ver gráfica | <input type="checkbox"/> | |
| <input type="checkbox"/> | ehIMc1s7KT | test | 2019-10-16 21:31:22 | 00:00:34 | 55.8 | 78.12 | Ver gráfica | <input type="checkbox"/> | |
| (Sin pesos) | a9PSHhDkbr | tuesday | 2020-06-09 18:31:51 | 01:16:30 | 1073 | 1073 | (Sin pesos) | <input type="checkbox"/> | |
| <input checked="" type="checkbox"/> | UDaUHvzMye | test | 2019-11-06 16:17:44 | 00:26:38 | 8.55 | 1.71 | Ver gráfica | <input type="checkbox"/> | |
| <input type="checkbox"/> | S3cQCqyeJc | parentsday1 | 2020-09-26 20:31:05 | 00:15:42 | 2.67 | 24 | Ver gráfica | <input type="checkbox"/> | |
| (Sin pesos) | QOnLyVtLAC | hello | 2020-02-24 16:16:50 | 01:12:39 | 27.66 | 19.36 | (Sin pesos) | <input type="checkbox"/> | |
| <input checked="" type="checkbox"/> | JmJ2por3Xk | parentsday3 | 2020-09-26 21:09:57 | 00:05:16 | 26 | 156 | Ver gráfica | <input type="checkbox"/> | |
| (Sin pesos) | i37RdnKRTS | test | 2019-11-06 16:58:52 | 01:04:24 | 32.8 | 19.68 | (Sin pesos) | <input type="checkbox"/> | |
| <input type="checkbox"/> | GihlPyhlFh | hello2 | 2020-02-24 18:58:56 | 00:16:06 | 7.14 | 35.7 | Ver gráfica | <input type="checkbox"/> | |
| (Sin pesos) | ENw2qHLuJG | test2 | 2020-09-22 15:15:50 | 00:38:54 | 27.7 | 24.93 | (Sin pesos) | <input type="checkbox"/> | |
| <input type="checkbox"/> | AHwGDvXm1Y | parentsday2 | 2020-09-26 20:57:10 | 00:11:56 | 3.53 | 24.7 | Ver gráfica | <input type="checkbox"/> | |

Tabla con todos los registros

Ilustración 90. Uso: ver múltiples registros

Con ello, se muestra una gráfica con cada nombre del registro seleccionado en la leyenda, y para cada serie su peso en gramos (eje y) e instante de tiempo (eje x) (ilustración 91).

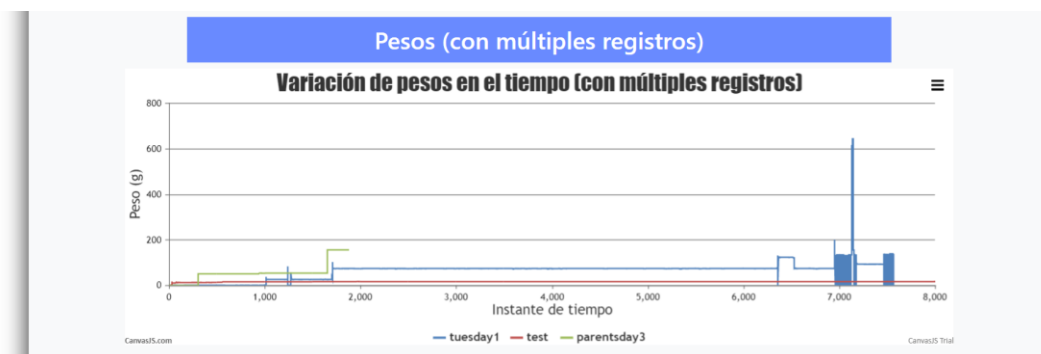


Ilustración 91. Uso: gráfica con múltiples registros

La penúltima columna *Ver gráfica* permite **ver las gráficas de un único registro** si se pulsa sobre el botón *Ver gráfica*. Al igual que la primera columna, en algunas ocasiones no existe la opción de ver las gráficas al no haber registro de los pesos.

Al pulsar el botón, se muestra de forma predeterminada el primer **gráfico Instante por peso** con los pesos en gramos (eje y) y los instantes de tiempo (eje x). Para cambiar a otro gráfico, se debe seleccionar otra opción en el menú desplegable y posteriormente pulsar en el botón *Cambiar gráfica* (ilustración 92).



Ilustración 92. Uso: gráfica de instante por peso

La segunda opción muestra el **gráfico Instante por peso diferente** con los mismos ejes, pero sin mostrar los pesos consecutivos que hayan sido iguales (ilustración 93).



Ilustración 93. Uso: gráfica de instante por peso diferente

La tercera opción muestra el **gráfico Instante por peso diferente (parte entera)**, que es lo mismo que el gráfico anterior, pero sin mostrar los pesos consecutivos que hayan sido iguales en su parte entera (ilustración 94).



Ilustración 94. Uso: gráfica de instante por peso diferente (parte entera)

De la cuarta a la séptima opción, se muestran los gráficos Peso medio en franjas de 10/20/50/100 instantes, que toman 10, 20, 50 o 100 pesos del total de

pesos haciendo la media de las 10, 20, 50 o 100 franjas de pesos (ilustraciones 95 a 98).



Ilustración 95. Uso: gráfica de peso medio en franjas de 10 instantes



Ilustración 96. Uso: gráfica de peso medio en franjas de 20 instantes



Ilustración 97. Uso: gráfica de peso medio en franjas de 50 instantes

Aplicación web para la visualización de datos de sensor “Smart Bin”



Ilustración 98. Uso: gráfica de peso medio en franjas de 100 instantes

La octava opción permite generar una **predicción con Prophet**, tomando de base una gráfica suavizada, aquella que tomaba el peso medio por cada segundo (serie azul), y así generar una predicción sobre los siguientes pesos (serie roja) (30% del total). A diferencia de las anteriores gráficas, esta tiene como eje x una fecha estimada en la que se tomó el registro de cada peso. Al querer hacer una predicción, las fechas de toma resultan más útiles que los instantes de tiempo. Por otro lado, aparece una tabla con todas las métricas que indican la precisión de la predicción. Para tener más claro que revelaba cada métrica, esto se remarcaba en el punto 6.6.2. *Sprint 5 Desarrollo* (ilustración 99).



Ilustración 99. Uso: gráfica de predicción con Prophet

La última opción permite generar una **predicción con Prophet con ajuste de Curva**, mostrando (1) la gráfica predeterminada (serie azul, parecida a la de la primera opción de todas, pero con fecha), (2) la gráfica con el ajuste de curva (serie verde) y (3) la gráfica que toma una predicción tomando como referencia el ajuste de curva (serie roja). Al igual que con la anterior gráfica, se

toman las fechas en vez de los instantes de tiempo y se muestran las métricas de la predicción (ilustración 100).



Ilustración 100. Uso: gráfica de predicción con Prophet con ajuste de curva

10.2.3. VENTANA DE VISTAS

En primer lugar, en la parte superior de la ventana de *Vistas* aparecen cuatro cuadros. Estos relejan los siguientes **datos generales**: (1) el número total de registros almacenados, (2) el número de registros que no incluyen tomas de pesos, (3) el peso más alto registrado en gramos y (4) el peso más bajo registrado en gramos que no sea 0 (ilustración 101).

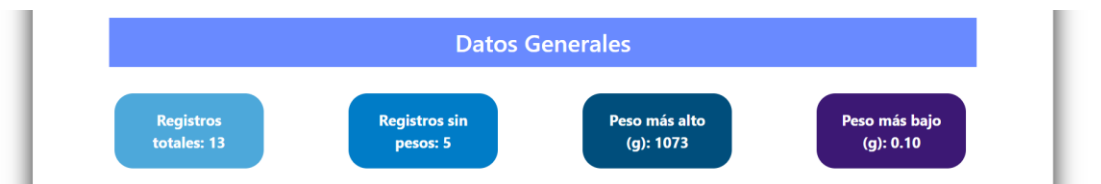


Ilustración 101. . Uso: datos generales

En segundo lugar, aparecen dos **gráficos sobre los pesos medios**, uno con los pesos medios por registro, y otro con los pesos medios por fecha de registro (ilustración 102).



Ilustración 102. Uso: gráficas con pesos medios

Por último, esta ventana ofrece los mismos dos gráficos anteriores, pero esta vez con los **pesos máximos** (ilustración 103).



Ilustración 103. Uso: gráficas con pesos máximos

10.2.4. VENTANA DE CONFIGURACIÓN

La última ventana ofrece una **configuración sobre los gráficos** con las siguientes opciones: (1) habilitar o deshabilitar el zoom (iconos de la lupa), (2) habilitar o deshabilitar el modo oscuro (icono del sol y la luna) y (3) habilitar o deshabilitar la exportación de los gráficos (iconos de descarga) (ilustración 104).

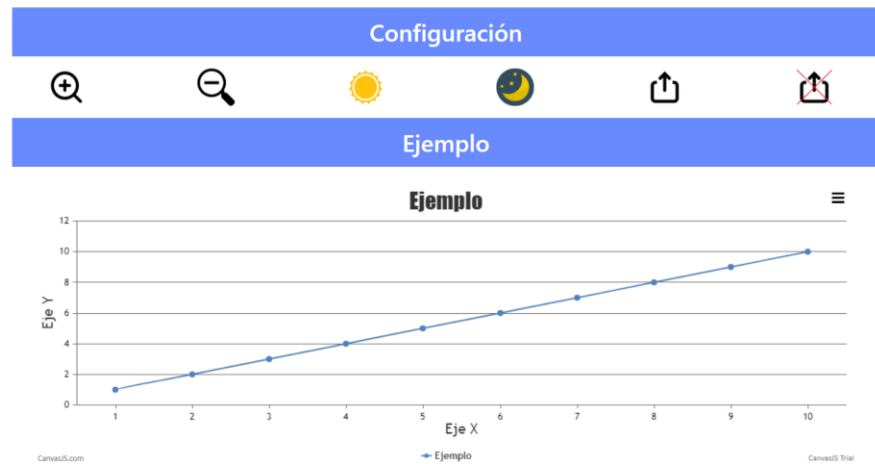


Ilustración 104: Uso: configuración de gráficas

Por último, cabe destacar como se usa el zoom o la opción de exportación. Para usar el zoom, se debe arrastrar el ratón dentro del gráfico haciendo una selección cuadrada. Y para recuperar el zoom original, se pulsa el icono con una flecha rotatoria en la parte superior derecha del gráfico (ilustraciones 105 y 106).



Ilustración 105. Uso: zoom 1



Ilustración 106. Uso: zoom 2

Finalmente, para usar el modo exportar, se debe seleccionar el botón con tres rayas horizontales que aparece en la parte superior derecha del gráfico (ilustración 107).



Ilustración 107. Uso: exportación

Para mayor información, puede acceder al siguiente vídeo explicativo:
https://www.youtube.com/watch?v=Z_XEh-Zz06s