UNIVERSIDAD DE BURGOS
ESCUELA POLITÉCNICA SUPERIOR
Grado en Ingeniería Informática

TFG del Grado en Ingeniería Informática

Estudio de métodos de selección de instancias en aprendizaje supervisado Documentación Técnica

Presentado por Daniel Puente Ramírez
en Universidad de Burgos — 11 de marzo de 2022
Tutor: Alvar Arnaiz González

Índice general

Índice de figuras

Índice de tablas

Apéndice A

Plan de Proyecto Software

A.1. Introducción

En este anexo se tratará el plan de proyecto, es la base sobre la que se crea el proyecto. Desde el punto de vista de la temporalidad y la viabilidad. Es una parte fundamental del ya que permitirá visualizar el escenario en el que se desarrollará el proyecto, permitiendo hacer una alineación estratégica de todos los elementos que se deben completar para finalizar correctamente el proyecto.

Desde el punto de vista de la planificación temporal, el proyecto sigue la metodología ágil *Scrum*. Permitiendo definir cada uno de los objetivos que se desean alcanzar, los elementos que los componen y su respectiva prioridad.

Scrum, de manera muy resumida, trabaja con u *product backlog*, es una lista de prioridades en función del valor de cada tarea. Cuando comienza un *sprint*, se empieza a trabajar en las tareas que se encuentren en el *sprint backlog*, estas han sido extraídas del *product backlog*. En el caso de este proyecto se realiza una reunión de planificación, *sprint planning*, cada dos semanas aproximadamente.

Para el control y seguimiento se utiliza una herramienta externa, *Zenhub*, la cual permite la definición de las tareas, el seguimiento de cada una de ellas en función de la planificación póker, seguimiento de cada *sprint*, el versionado, etc.

Seguidamente se realizará un estudio de la viabilidad del proyecto, tanto a nivel económico como legal.

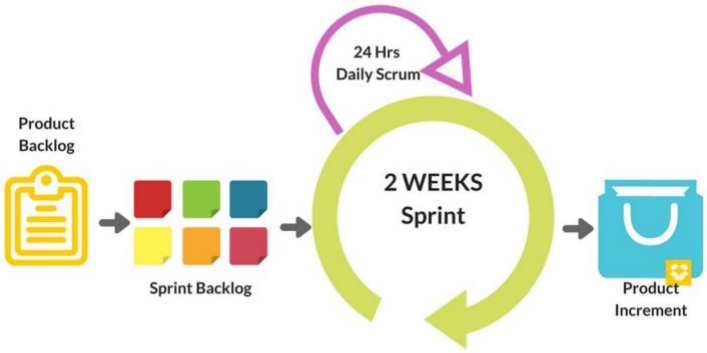

Figura A.1: Metodología *scrum*.

A.2. Planificación temporal

SCRUM

Scrum es un marco de trabajo que permite el trabajo colaborativo en equipos. Permite que los equipos que trabajan en proyectos con esta metodología se organicen por sí mismos, siendo ellos los que deciden cómo afrontar los problemas que van surgiendo.

Según [4], el modelo *Scrum* se basa en tres componentes principales: roles, procesos y artefactos. El *Scrum Master* es el puesto asumido por el director o gerente del proyecto, o en algunos casos el líder del equipo. Esta figura representa los valores y principios por los que se rige la metodología de *scrum*, manteniendo los valores y buenas prácticas, así como resolviendo los impedimentos que vayan surgiendo a lo largo del desarrollo del proyecto. Habitualmente los equipos están compuestos por entre cinco y diez personas que trabajan en el proyecto a tiempo completo. Siendo este equipo independiente y flexible en cuanto a jerarquía interna, no siendo representado el papel del "jefe" dentro de este por la misma persona siempre. Esto genera que el papel cambie en función de las necesidades del propio proyecto, la configuración del equipo cambia únicamente entre iteraciones, o *sprints*, no dentro de los mismos.

Sprints

Los *sprints* son periodos breves de **tiempo fijo** en el que el equipo trabaja para completar una cantidad de trabajo pre-establecida. Si bien muchas guías asocian los *sprints* a la metodología ágil, asociando la metodología ágil y la metodología seguida en *scrum* como si fueran lo mismo, cuando no lo son. La metodología ágil constituye una serie de principios, y la metodología *scrum* es un marco de trabajo con la única finalidad de conseguir resultados.

A pesar de las similitudes los *sprints* poseen un objetivo subyacente, entregar con frecuencia *software* de trabajo.

Sprint meetings

Dentro de la metodología *scrum* existen diferentes reuniones que favorecen la agilidad del proyecto y que todo el mundo sepa lo que tiene que hacer en cada momento.

- ***Sprint planning meeting.*** Esta reunión puede tener una duración de hasta de un día completo de trabajo. En ella deben de estar presentes todas las partes del proyecto, i.e. el *Scrum Master*, el equipo de desarrollo, y el *product owner*. Poseen dos partes, en la primera de ellas se define el *product backlog*, requerimientos del proyecto y se definen los objetivos para el *sprint* que comienza, i.e. lo que se espera "construir" o completar en el *sprint*. En la segunda parte de la reunión se trabaja en el *sprint backlog*, las tareas que se van a seguir en el *sprint* para completar el objetivo de éste.
- ***Daily meeting.*** Debido a que los requerimientos del proyecto no se pueden variar durante la vida de un *sprint*, existen las reuniones diarias que son organizadas por el *Scrum Master* en las que se comenta el trabajo del día previo, lo que se espera de ese día y qué está retrasando o impidiendo a un individuo el proseguir con sus tareas, esta reunión no debe tener una duración de más de quince minutos y se debe realizar "de pie". No es una reunión para ver quién retrasa el proyecto sino para ayudar a quién lo necesite entre todos los miembros del equipo y permitir esa agilidad.
- ***Sprint review meeting.*** Reunión fijada al final de cada *sprint* en la cual se hace una puesta en conocimiento de lo que se ha realizado en ese *sprint*, siempre que se pueda se hará una demostración funcional en lugar de una presentación al *product owner*. Esta reunión tiene un carácter informal.

Artifacts

Uno de los componentes más importantes de cara a la metodología *scrum* son los artefactos, o *artifacts* por su nombre en inglés. Éstos incluyen el *product backlog*, el *sprint backlog* y los *burn down charts*.

- ***Product backlog.*** Lista de trabajo ordenada por las prioridades para el equipo de desarrollo. Es generada a partir de las reuniones de planificación de los *sprints*, contiene los requisitos. Se encuentra actualizado y clasificado en función de la periodicidad asignada a las tareas, pudiendo ser de corto o largo plazo. Aquellas tareas que se deban resolver a corto plazo deberán estar perfectaemnte descritas antes de asignarlas esta periodicidad, implicnddo que se han diseñado las historias de usuario completas así como el equipo de desarrollo ha establecido las estimaciones correspondientes. Los elementos a largo plazo pueden ser abstractos u opacos, conviene que estén estimados en la medida de lo posible para poder tener en cuenta el tiempo que llevará desarrollarla.

  Los propietarios del producto dictan la prioridad de los elementos de trabajo en el *product backlog*, mientras que el equipo de desarrollo dicta la velocidad a la que se trabaja en *backlog* [11]

  La estimación es una parte muy importante ya que es lo que permitirá al equipo de desarrollo mantener el ánimo y el trabajo al ritmo deseado. La estimación es realizada en la *sprint planning meeting*, en la que se estima para cada tarea/producto del *product backlog*. No se busca tener un resultado exacto del tiempo que va a llevar al equipo completar esa tarea, sino es una previsión. Para realizar correctamente la estimación se debe tener en cuenta el tamaño y la categoría de la tarea, los puntos de historia que se le van a asignar, así como el número de horas y días que van a ser necesarias para completar la tarea.

- ***Sprint backlog.*** Lista de tareas extraídas del *product backlog* que se han acordado desarrollarse a lo largo de un *sprint*. Este *backlog* es seleccionado por el propio equipo de desarrollo, para ello seleccionan una tarea del *product backlog* y se divide en tareas de menor tamaño y abordables. Aquellas tareas de menor tamaño que el equipo no haya sido capaz de desarrollar previo a la finalización del *sprint* quedarán almacenadas para próximos *sprints* en el *sprint backlog*.

Actores, roles y responsabilidades

Dentro de un equipo que sigue la metodología *scrum* encontramos diferentes actores, como ya se ha comentado el equipo de desarrollo suele estar compuesto por entre cinco y diez personas, además del *Scrum Master* y el *Product Owner* [12]

- ***Product Owner.*** Encargado de optimizar y maximizar el valor del producto, es la persona encargada de gestionar las prioridades del *product backlog*. Una de sus principales tareas es la de intermediario con los *stakeholders*, partes interesadas, del proyecto; junto con recoger los requerimientos de los clientes. Es habitual que esta figura sea representante del negocio, con lo que aumenta su valor.

  Para cada *sprint* debe de marcar el objetivo de éste de manera clara y acordada con el equipo de desarrollo, lo cual hará que el producto vaya incrementando constantemente su valor. Para que todo fluya como debe, esta figura tiene que tener el "poder" de tomar decisiones que afecten al producto.

- ***Scrum Master.*** Figura con dos responsabilidades, gestionar el proceso *scrum* y ayudar a eliminar impedimentos que puedan afectar a la entrega del producto.

  1. Gestionar el proceso *scrum*. Su función es asegurarse de que el proceso se lleva a cabo correctamente, facilitando la ejecución de éste y sus mecánicas. Consiguiendo que la metodología sea una fuente de generación de valor.
  2. Eliminar impedimentos. Eliminar los problemas que vayan surgiendo a lo largo de los *sprints* con el fin de mantener el ritmo de trabajo dentro de los equipos de desarrollo para poder entregar valor, manteniendo la integridad de la metodología.

- **Equipo de desarrollo.** Formado por entre cinco y diez personas encargados del desarrollo del producto, organizados de forma autónoma para conseguir entregar las tareas del *product backlog* asignadas al *sprint* correspondiente. Para que funcione correctamente la metodología todos los integrantes deben de conocer su rol dentro del equipo, internamente se pueden gestionar como el equipo considere, pero de cara "hacia fuera" son un equipo con una responsabilidad.

Planificación por *sprints*

La organización temporal del proyecto se ha organizado siguiendo los estándares de la metodología *scrum*, i.e. usando *sprints*.

Inicialmente la *sprint planning meeting* es realizada cada dos semanas, debido a una falta de costumbre de trabajo con esta metodología se combina junto con la *sprint review meeting*, de forma que en una sola reunión se comenta tanto lo que se ha hecho como lo que está por realizarse en el siguiente *sprint*.

La velocidad de desarrollo del proyecto es una incógnita, debido a la no existencia de referencias previas del equipo de desarrollo del proyecto, en proyectos de ésta índole. Por lo tanto, la duración de los *sprints* puede que se vea ajustada a lo largo de la vida del proyecto.

No se utilizan *daily meetings* puesto que a pesar de que se invierte una media de tres a cinco horas diarias en el desarrollo, no es considerada necesaria. Si bien en caso de problemas se acuerda una reunión para el día siguiente con el fin de mantener la agilidad y no retrasar el proyecto.

Sprint 0: *Lights out and away we go!*

El *sprint* con el que comienza el desarrollo de este proyecto no ha seguido la metodología *scrum*, puesto que se formuló desde un punto de vista de toma de contacto inicial con el trabajo de investigación y todo lo que ello conlleva.

Los objetivos definidos han sido:

1. Lectura de *papers* relacionados con el ámbito de la inteligencia artificial. En concreto *SSL density peaks* [14], *Co-Training* [2], *Tri-Training* [17] y *Democratic Co-Learning* [16].

El tiempo empleado en la lectura y asimilación de estos conceptos ha sido de catorce horas, es la primera vez que se leen *papers* o artículos científicos completos procurando asimilar todos los conceptos de éstos. Se ha desarrollado entre el veintisiete de octubre y el cinco de noviembre, de dos mil veintiuno.

Sprint 1: Chad

- ***Planning meeting***

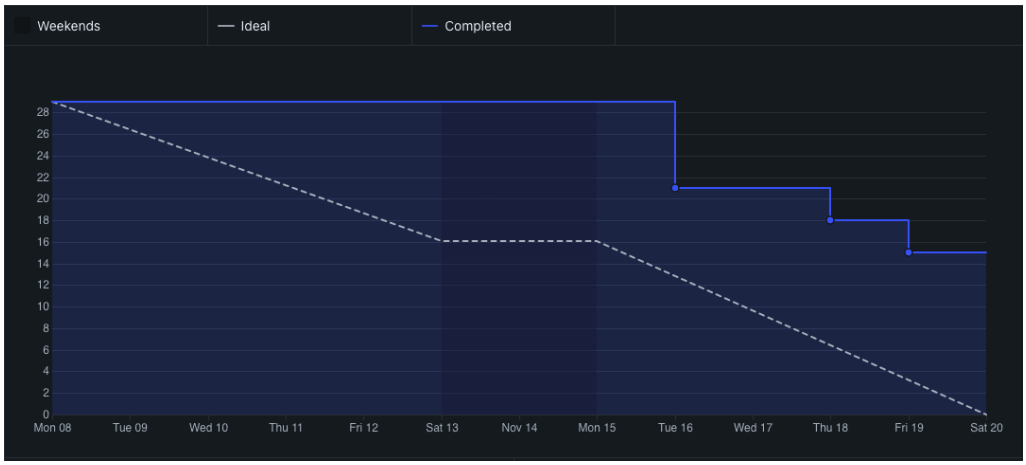

Figura A.2: *Burndown Chart Sprint 1.*

Objetivos del primer *sprint*:

1. Lectura del API de *scikit-learn*. Comprensión del funcionamiento de los transformadores y estimadores enfocado desde el punto de su programación.
2. Lectura de los *papers On issues instance selection* [10], *Comparison of instances seletion algorithms I. algorithms survey* [8] y *Comparison of instance selection algorithms II. Results and comments* [6].
3. Implementación de las técnicas de reducción del conjunto de entrenamiento, basados en k-NN.

- **Marcas temporales** El *sprint* se desarrolla entre el ocho y el diecinueve de noviembre de dos mil veintiuno. Han sido dedicadas al desarrollo del proyecto treinta horas.

- ***Burndown chart*** Durante este *sprint* el trabajo inicial comenzó ligeramente retrasado, motivos en el apartado *sprint review meeting*, por lo tanto podemos observar en la Figura A.2 como el trabajo completado dista del ideal o proyectado para este *sprint*.

  En el *sprint backlog* habían sido incluidos todos los algoritmos a programar, es por ello que indica que se ha completado aproximadamente la mitad del trabajo.

- ***Sprint review meeting*** El trabajo en este primer *sprint* ha salido adelante correctamente. Al ser el primer *sprint* ha habido un pequeño error de configuración del repositorio junto con la herramienta ZenHub, de ahí que en el *burndown chart* de esta semana, Figura A.2, aparezca como que la primera semana del *sprint* no ha habido trabajo completado.

  La adaptación a la metodología ágil ha resultado un poco compleja.

Sprint 2: Holleyman

- ***Planning meeting***

  Objetivos del segundo *sprint*:

  1. Finalizar implementación de los algoritmos basados en técnicas de reducción del conjunto de entrenamiento.
  2. Añadir la documentación correspondiente a los algoritmos implementados.
  3. Comprobar el rendimiento de los algoritmos implementados respecto a los resultados de una ejecución similar con el software *Weka*.

  El *sprint* se desarrolla entre el veintidós de noviembre y el tres de diciembre de dos mil veintiuno. Han sido dedicadas al desarrollo del proyecto treinta y ocho horas.

- ***Burndown chart***

  El trabajo realizado a lo largo de este *sprint* ya ha sido adecuado a la metodología *scrum*, obteniendo un *burndown chart*, Figura A.3, con más sentido que la que se había obtenido en el *Sprint* 1.

  El equipo de desarrollo se sigue habituando poco a poco a la metodología de trabajo y en este *sprint* se ha trabajo por debajo del "ideal" para el proyecto.

- ***Sprint review meeting***

  A lo largo de este *sprint* se descubrió un problema en la forma de identificar los k-NN en el algoritmo *Condensed Nearest Neighbor, CNN* [7], retrasando el trabajo cuatro horas, entre identificación y re-programación. Este error se descubrió mientras se investigaba otro error, en este caso el algoritmo *Iterative Case Filtering, ICF* [3] terminaba en error buscando los k-NN de las últimas instancias.

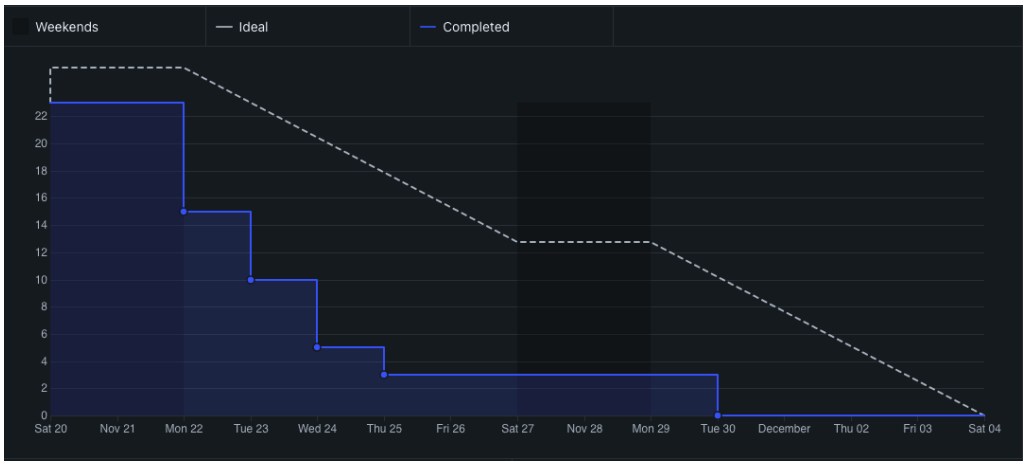

Figura A.3: *Burndown Chart Sprint 2.*

La implementación de los algoritmos *Reduced Nearest Neighbor, RNN* [5] y *Modified Selective Subset, MSS* [1] ha sido relativamente asequible una vez se comprendía el algoritmo en cuestión así como su funcionamiento (entradas, procesado, salidas...).

***Sprint* 3: Manion**

- ***Planning meeting***

  Objetivos del tercer *sprint*:

  1. Comenzar la documentación del proyecto.
     - Comenzar la memoria por el marco teórico.
     - Comenzar los anexos por la planificación temporal.

     Se va a realizar en LaTeX.
  2. Aprender lo básico de LaTeXlo más rápido posible para poder trabajar con él.
  3. Buscar la precisión de los algoritmos implementados con conjuntos etiquetados de $[1\,\%, 5\,\%, 10\,\%, 20\,\%, 40\,\%, 60\,\%, 80\,\%, 100\,\%]$ del conjunto total. En búsqueda de las asíntotas donde ya no mejora la clasificación.
  4. Validación de los algoritmos de selección de instancias con *Weka* y *KNN*.

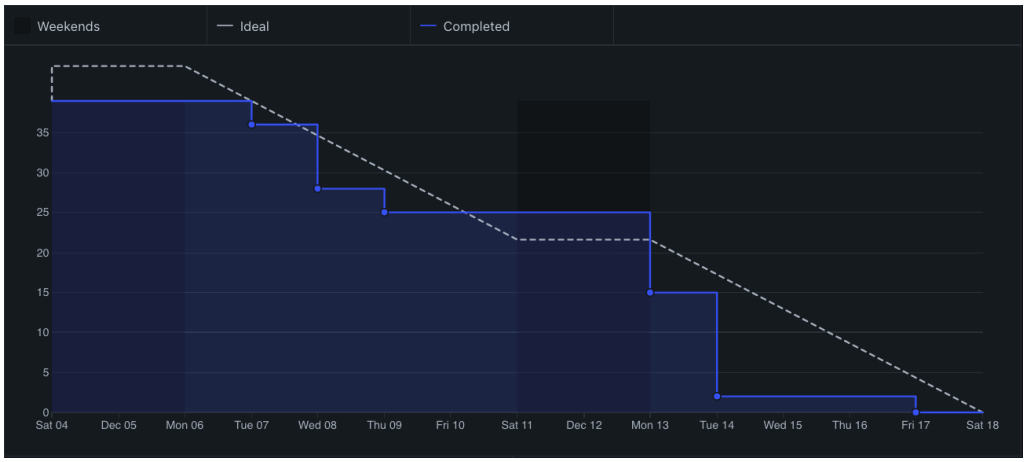

Figura A.4: *Burndown Chart Sprint 3.*

- **Marcas temporales**

  El *sprint* se desarrolla entre el seis y el diecisiete de diciembre de dos mil veintiuno. Han sido dedicadas al desarrollo del proyecto 45 horas.

- ***Burndown chart***

  El trabajo realizado en este tercer *sprint* ha sido realizado a un ritmo constante y con una dedicación en número de horas un poco mayor a los anteriores, como se puede ver en el *Burndown report*, ver figura A.4, el número de *story points* de este *sprint* era de 39 y a pesar de que algunas tareas llevaron más tiempo del inicialmente planificado, otras resultaron ser totalmente lo contrario, mucho más rápidas de realizar.

- ***Sprint review meeting***

  Este *sprint* ha sido un poco más grande en cuanto a horas de trabajo invertidas ya que el tiempo del equipo de desarrollo así lo ha permitido. A su vez se han detectado *bugs* en la codificación de algoritmos como ICF (se arreglará en el siguiente *sprint*) el cual después de comparar sus resultados contra los expresados por *Weka* con 9 conjuntos de datos no considerados como de juguete, la codificación del proyecto obtiene soluciones 20 % inferiores; el resto de los algoritmos implementados están en el rango de ±2 %. A su vez también se han descubierto limitaciones de otros algoritmos como es el caso de ENN cuando tiene pocas muestras y un número elevado de clases diferentes.

Se ha proseguido con la redacción de la memoria del trabajo, finalizando la primera parte de conceptos teóricos y comenzando la explicación teórica de los algoritmos que

***Sprint* 4: The Seven**

- ***Planning meeting***

  Objetivos del cuarto *sprint*

  1. Revisar y corregir la codificación del algoritmo *Iterative Case Filtering*, *ICF*.
  2. Formatear las métricas de rendimiento recogidas durante el *sprint* anterior.

- **Marcas temporales**

  El *sprint* se desarrolla entre el dieciocho y el veintitrés de diciembre de dos mil veintiuno. Han sido dedicadas al desarrollo del proyecto 21 horas. Este *sprint* posee una duración más corta con el fin de ajustar las reuniones a las festividades propias de la Navidad.

- ***Burndown chart***

  En el *Burndown report* asociado a este *sprint*, ver figura A.5, se aprecia como el trabajo ha sido finalizado en unas marcas temporales muy por delante de lo «ideal». Esto se debe a que el equipo de desarrolló comenzó a realizar el trabajo el sábado 18 de diciembre, en lugar de esperar al lunes 20, bajo la presunción de que el trabajo asignado iba a ser mayor.

- ***Sprint review meeting*** Ha pesar de la corta duración del *sprint* para poder organizar el siguiente *sprint* antes de las festividades navideñas, el trabajo realizado ha sido correcto e importante, debido a que para poder seguir trabajando en otros algoritmos de selección de instancias o de aprendizaje semi-supervisado, lo anterior debe de quedar correctamente hecho. Es por ello, que prácticamente se le ha dedicado un *sprint* entero a arreglar el algoritmo *Iterative Case Filtering, ICF*, ya que con él y a falta de implementar DROP3, ya tendríamos todos los algoritmos de selección de instancias correctamente implementados.

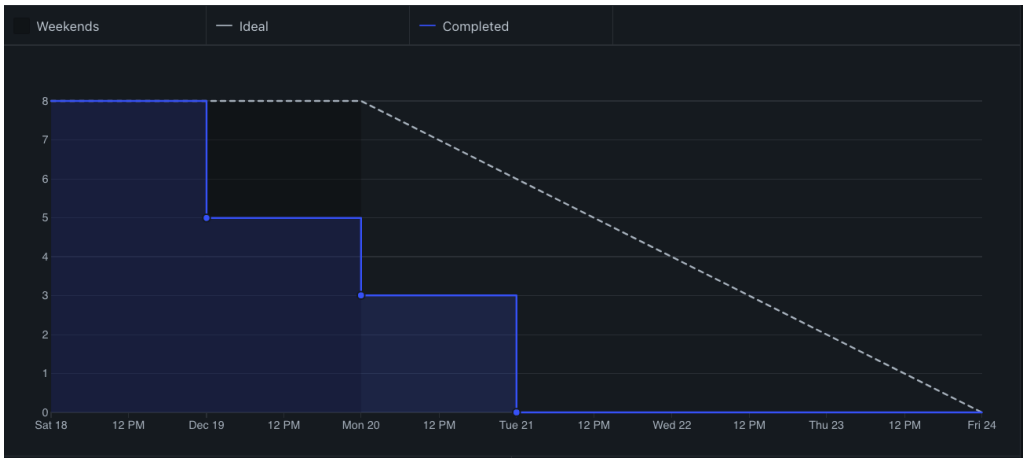

Figura A.5: *Burndown Chart Sprint 4.*

Sprint 5: Murph

- ***Planning meeting***

  Objetivos del quinto *sprint*:

  1. Codificación de los algoritmos de aprendizaje semi-supervisado: Co-Training [2], Tri-Training [17], Democratic Co-Learning [16] Y del algoritmo de selección de instancias DROP3 [13]
  2. Implementar los algoritmos anteriores como clases para poder ser utilizados con métodos *fit* y *predict*.
  3. Escribir la documentación de la memoria referente a los anteriores algoritmos.
  4. Escribir la planificación temporal relativa a los *sprints* 4 y 5.
  5. Añadir leyenda a la figuras generadas con *self-training* en función de un % de datos etiquetados.

  Se espera que sea un *sprint* muy productivo debido a las fechas en las que se realiza y la mayor disponibilidad del equipo de desarrollo.

- **Marcas temporales**

  Este *sprint* se desarrolla entre el veinticuatro de diciembre de dos mil veintiuno y el diez de enero de dos mil veintidós. Tiene una duración igual a las festividades navideñas.

Figura A.6: *Burndown Chart Sprint 5.*

- ***Burndown chart***

  En este *sprint*, y como vemos en la figura A.6, referente al correspondiente *Burndown report*; se ha realizado una gran cantidad de trabajo, habiendo siendo completados 77 puntos de historia, una cantidad muy superior a anteriores *sprints*, esto es debido en gran parte a las fechas en las que nos encontramos, ya que el número de horas que se han podido invertir en el desarrollo del proyecto ha sido muy superior a lo que venían siendo habituales. En total han sido utilizadas cerca de 110 horas de trabajo, siendo repartidas en los 17 días que ha durado el *sprint* y con una media de horas de trabajo de 6.5 horas diarias.

- ***Sprint review meeting***

  Todo el trabajo que se ha realizado en este *sprint* podría haber sido realizado seguramente en tres cuartas partes del tiempo real invertido, pero debido al tiempo de lectura de los artículos de donde se extraían los algoritmos, así como su correcta comprensión, codificación y posterior resolución de problemas asociados a *bugs* que se van descubriendo «sobre la marcha», ha sido un *sprint* largo y en algunos momentos agotador.

  A falta de realizar las correspondientes pruebas de validación de los algoritmos implementados, para comprobar que son correctas las implementaciones, ya se encontrarían finalizados todos los algoritmos de selección de instancias.

Sprint 6: Bert

- ***Planning meeting***

  Objetivos del sexto *sprint*:

  1. Verificación de la correcta implementación del algoritmo de selección de instancias DROP3. Se realizará como se ha venido trabajando anteriormente, contra los resultados propuestos para la misma parametrización, por Weka.
  2. Verificación de la correcta implementación de los algoritmos de aprendizaje semi-supervisado: *Co-Training*, *Tri-Training* y *Democratic Co-Learning*.
  3. Comenzar a escribir las secciones de «Técnicas y herramientas» y «Trabajos relacionados».

- **Marcas temporales**

  Este es un *sprint* relativamente corto, puesto que es de verificación de que el trabajo realizado hasta el momento es correcto, antes de pasar a otro «bloque» de trabajo. Comienza el martes once de enero de dos mil veintidós, y finaliza el lunes diecisiete de enero de dos mil veintidós.

- ***Burndown chart***

  Tal y como se aprecia en la Figura A.7 referente al sexto *sprint*, el ritmo de trabajo ha sido constante a lo largo de la primera semana, torciéndose al final del *sprint* debido a la complejidad sobrellevada de aprender la librería *Flask* y sus dependencias. Es por ello que dos *issues* se cerraron un día más tarde de la planificación. El número de horas aproximado que se han invertido han sido de 50h, permitido en gran medida con que todavía no hay clases del segundo cuatrimestre.

- ***Sprint review meeting***

  El trabajo realizado en este *sprint* ha ido de acuerdo a lo que se comentó en la anterior reunión. Si bien ha sido un *sprint* más enfocado a «cerrar» una parte de trabajo que se llevaba realizada para poder comenzar con el mejor pie posible la segunda etapa.

  Un punto de inflexión realizado en este *sprint* ha sido la refactorización de gran parte del repositorio, dejándolo en un formato de paquetes.

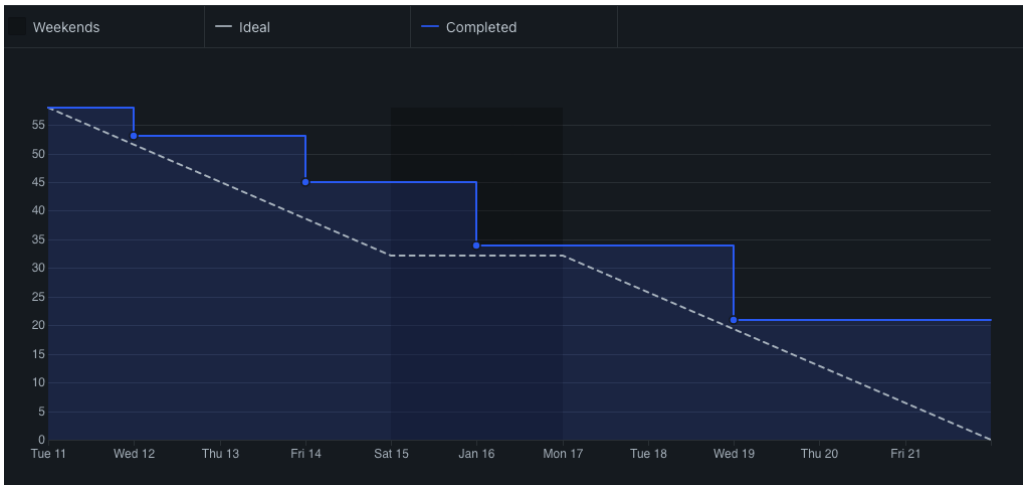

Figura A.7: *Burndown Chart Sprint 6.*

***Sprint* 7: Felix The Cat**

- ***Planning meeting***

  Objetivos del séptimo *sprint*:

  1. Modificar el algoritmo ENN para poder utilizarlo con Semi-Supervisado según el método de borrado de instancias.
  2. Preparar *scripts* para la experimentación y posterior visualización de hipótesis.
  3. Modificar la memoria en función de los comentarios de Alvar.
  4. Añadir a los trabajos relacionados UBUMLaaS. Aunque es parte del propio Trabajo de Fin de Grado, no deja de ser una herramienta de MLaaS.
  5. Añadir *Self Training* a UBUMLaaS.

- **Marcas temporales**

  Este *sprint* se desarrolla entre el veinticinco de enero de dos mil veintidós y el dos de febrero de dos mil veintidós. Es un *sprint* muy rápido de preparación para poder comenzar con la parte de UBUMLaaS.

- ***Burndown chart***

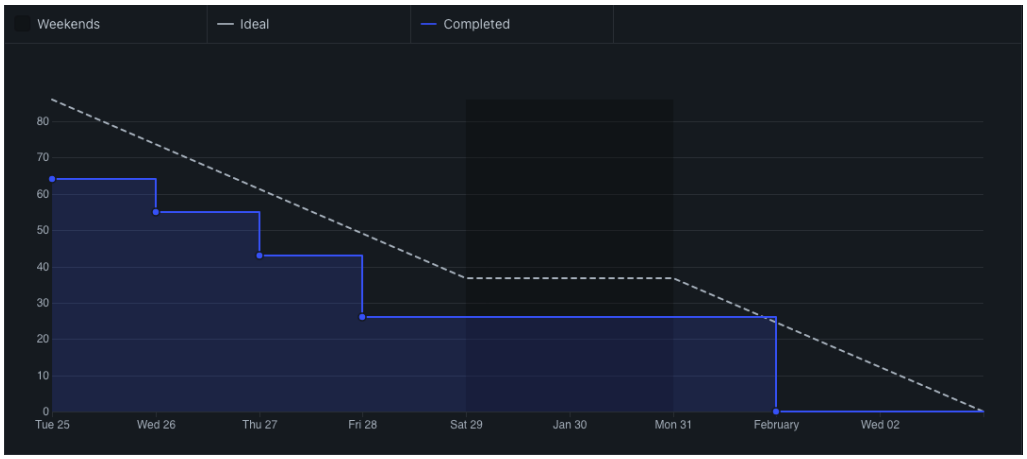

Figura A.8: *Burndown Chart Sprint 7.*

El *Burndown* de este *sprint* representa un ritmo de trabajo «adelantado» a la velocidad óptima, esto se debe a que como en el *sprint* anterior no se cerraron todas las *issues* previa la finalización del mismo, pero sí fueron cerradas previo el inicio de este nuevo *sprint*, el gráfico queda por debajo siempre. El número de horas invertido ha rondado las 35h. Un *sprint* de menor tamaño.

- ***Sprint review meeting***

  Este *sprint* si bien es como el anterior muy corto, y se ajusta a la temporarización del proyecto, ha tenido una carga de trabajo un poco más alta de lo esperado, esto se debe a que la integración de nuevos algoritmos a UBUMLaaS parecía en un primer momento muy directo, pero se han requerido hacer modificaciones con las que no se contaba en un primer momento.

Sprint 8: Name

- ***Planning meeting***

  Objetivos del octavo *sprint*:

  1. Crear una nueva selección en «Nuevo Experimento» en UBUMLaaS para los algoritmos de aprendizaje Semi-Supervisado.
  2. Integrar los algoritmos implementados de Semi-Supervisado en la plataforma UBUMLaaS.

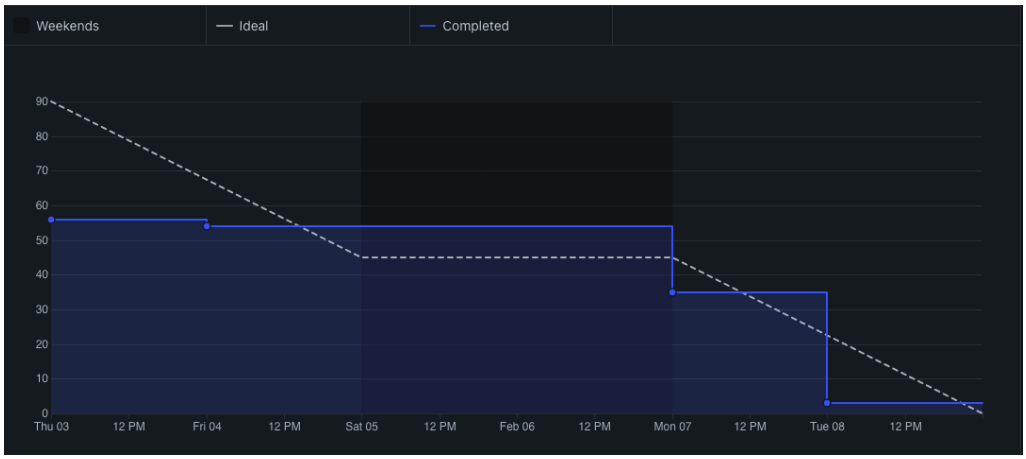

Figura A.9: *Burndown Chart Sprint 8.*

3. Comenzar a traducir parte de la interfaz como parte de un trabajo paralelo. (Puede que la versión final soporte varios idiomas, decisión de diseño aún por tomar.)
4. Crear los rankings con Python de las experimentaciones realizadas, principalmente de 3-NN sin borrado.
5. Hacer un *refactor* general al proyecto. Inicialmente tenía una estructura de carpetas, se desea una estructura de paquetes.
6. Hacer el proyecto accesible desde PIP[1].

- **Marcas temporales**

  Este *sprint* se desarrolla entre el tres de febrero de dos mil veintidós y el ocho de febrero de dos mil veintidós. Nos encontramos ante otro *sprint* ya con la nueva dinámica de trabajo, de duración aproximada a una semana.

- ***Burndown chart***

  Tal y como se aprecia en la Figura A.9 se aprecia que el ritmo de trabajo en este *sprint* ha sido muy escalonado, el número de *issues* no ha sido muy elevado, pero la complejidad de estas sí que lo ha sido, es por ello que se planificó 90 puntos de historia. Seguidamente podemos

[1]Sistema de gestión de paquetes utilizado para instalar y administrar paquetes de *software* escritos en Python.

apreciar como entre el cuatro y el siete de febrero, coincidiendo con el fin de semana, no ha habido trabajo finalizado, se debe a unas mini-vacaciones que se tomó el equipo de desarrollo.

- ***Sprint review meeting*** Con el *sprint* finalizado se ha visto como lo que parecía una planificación para una o dos semanas, ha quedado resuelta dentro del propio *sprint*. El equipo de desarrollo comienza a familiarizarse con el *backend* de UBUMLaaS propiciando un desarrollo más eficaz de las tareas que se van encomiando.

  Destacar que no se finalizan todas las tareas en tiempo, sino que se finaliza una en la noche del martes ocho, ya casi de madrugada, entrando técnicamente en el siguiente *sprint*.

Sprint 9: Name

- ***Planning meeting***

  Objetivos del noveno *sprint*:

  1. Continuar con la traducción del *frontend* en los «tiempos muertos», aún no se ha decidido si finalmente pasará a producción o no.
  2. Crear un nuevo conjunto de gráficas y relanzar experimentaciones con SVC como referencia. El método de eliminación de instancias con etiqueta conocida queda descartado, únicamente se trabajará para la experimentación con la aproximación que no las elimina.
  3. Crear los nuevos rankings basados en la precisión.
  4. Comprobar la implementación de los algoritmos *Co-Training*, *Tri-Training* y *Democratic Co-Learning* contra los implementados por Jose Luis Garrido Labrador (Investigador del grupo ADMIRABLE).
  5. Añadir a UBUMLaaS los filtros implementados en los anteriores *sprints*.

- **Marcas temporales**

  Este *sprint* se desarrolla entre el nueve de febrero de dos mil veintidós y el dieciséis de febrero de dos mil veintidós.

- ***Burndown chart***

  Este *sprint* tiene una duración de una semana, tal y como se desea (aproximadamente) que sean a partir de febrero. A este *sprint* se le

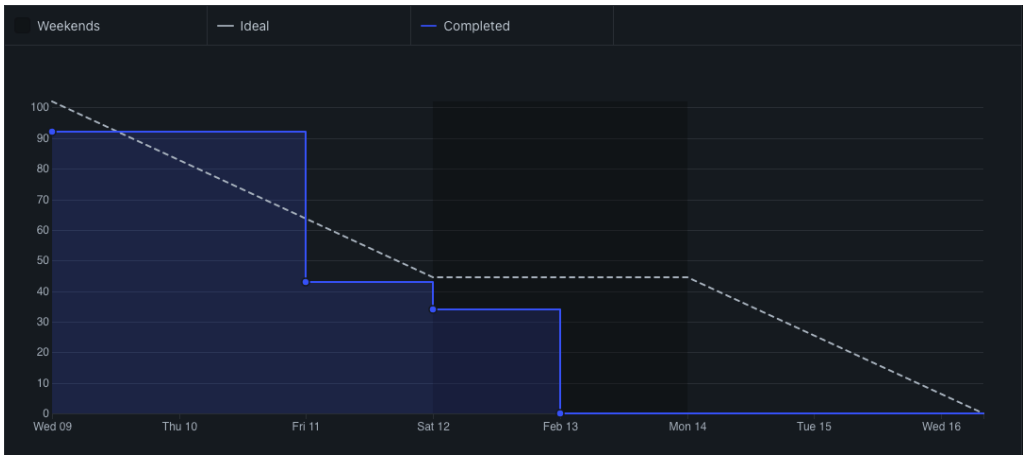

Figura A.10: *Burndown Chart Sprint 9.*

asignó una gran carga en cuanto a puntos de historia se refiere con un total de 102, las horas invertidas por el equipo de desarrollo no han llegado a las 55, hay una desviación de los puntos de historia y las horas invertidas, se comentará más adelante.

Quedando el trabajo finalizado un par de días antes de la fecha de finalización del *sprint*, dando al equipo de desarrollo tiempo para planear futuras tareas y aproximaciones a problemas conocidos.

- ***Sprint review meeting***

  En este *sprint* se planificó «tirando a lo alto» en los puntos de historia, se debe a que se requería comprobar la implementación de los tres algoritmos de aprendizaje Semi-Supervisado, y en caso de que alguno (o todos) tuviera una implementación incorrecta, realizar los ajustes pertinentes para que fuera correcta. Debido a la experiencia del equipo de desarrollo un mes atrás con el filtro ICF, el cual tuvo que ser re-programado y revisado en más de una ocasión debido a su inconsistencia, se aventuró un futuro similar con éstos algoritmos ya que son más grandes y con una complejidad superior. La realidad en este caso superó las expectativas del equipo de desarrollo, cuando los tres algoritmos tuvieron una desviación menor al 1 % en comparación con los de Jose Luis. (En futuros *sprints* se ha propuesto ser más críticos con la asignación de puntos de historia para no tener diferencias de este calibre).

Con todo y con ello, la implementación de los filtros en UBUMLaaS incurrió en múltiples modificaciones a la estructura base de la propia plataforma, pero con un resultado satisfactorio.

Los rankings creados no han convencido en estructura y formato, es por ello que en siguiente *sprint* tendrán que ser repetidos.

Sprint 10: Name

- ***Planning meeting*** Objetivos del décimo *sprint*:

  1. Rehacer las gráficas de rankings en la experimentación.
  2. Comenzar con la parte de administración de UBUMLaaS.
     a) Crear una nueva interfaz que de soporte a esta nueva funcionalidad que va a poseer la aplicación.
     b) Integrar nuevos campos en el registro de usuarios, tales como su país de origen y el uso deseado que se le va a dar a aplicación.
     c) Crear una interfaz de administración de usuarios (añadir usuarios, activarlos, hacerlos administradores o eliminarlos).
     d) Crear una primera interfaz básica de *dashboard analytics* del sistema.

- **Marcas temporales**

  Este *sprint* se desarrolla entre el dieciséis de febrero de dos mil veintidós y el veintiuno de febrero de dos mil veintidós.

- ***Burndown chart***

  Este *sprint* ha sido más ajustado el número de horas invertidas en el desarrollo de las tareas marcadas en comparación los puntos de historia. Se han marcado un total de 46 puntos de historia y se han invertido 35 horas de trabajo. Siguiendo un poco más la tónica de otros *sprints*. En los primeros días, tal y como se aprecia en la Figura A.11, sí que hubo *commits* pero no se cerraron tareas debido a que se trabajó en «paralelo» sobre varias *issues* a la vez, ya que toda la parte de crear la interfaz de administración y las páginas que la iban a comenzar a formar parte de la misma, se encuentran fuertemente inter-relacionadas.

- ***Sprint review meeting***

  El trabajo realizado durante este *sprint* ha sido más duro que el de *sprints* anteriores. Esto se debe a la poca experiencia del equipo de

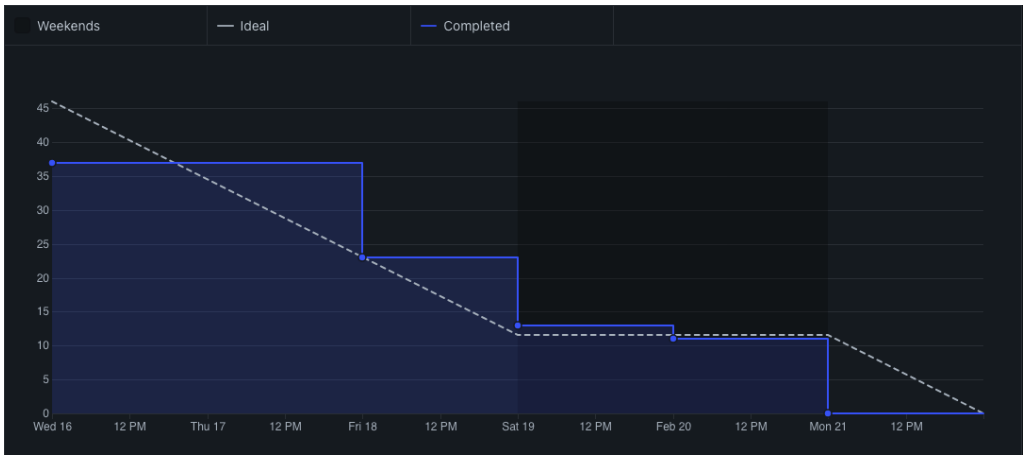

Figura A.11: *Burndown Chart Sprint 10.*

desarrollo con aplicaciones que poseen un *frontend*, el uso de JavaScript, jQuery, AJAX, ... es algo que hasta la fecha no se había utilizado en gran medida y ahora es con lo que más se está trabajando, entonces ha requerido de un esfuerzo extra.

La parte de administración de UBUMLaaS ha sido creada con una base más moderna, sencilla y clara. Siguiendo el esquema de colores de la Universidad de Burgos. Es por ello que ahora mismo parecen dos aplicaciones diferentes, (la parte de administración en comparación con la parte de funcionalidad de MLaaS propiamente dicha).

Durante la realización del *sprint* fueron surgiendo pequeños *bugs* en la interfaz gráfica que se fueron solventando, todos ellos originados por descuidos (debido a la falta de experiencia) del propio equipo de desarrollo con el uso de las nuevas librarías.

Sprint 11: Name

- ***Planning meeting***

  Objetivos del undécimo *sprint*:

  1. Con [9] se desea comprobar con 16 de los 18 conjuntos de datos utilizados en sus experimentos los resultados esperados para comprobar si merece la pena continuar la línea de investigación con el enfoque inicial.

2. Montar un servidor con Jenkins, se desea incorporar a UBUMLaaS y a la librería *IS_SSL* dentro de CI/CD [2]
3. Añadir al *dashboard analytics* gráficos de carta con el número de experimentos de cada tipo que se han ejecutado. Así como los tiempos de uso de cada algoritmo.
4. Crear una pantalla de carga para el *dashboard* de forma que la recuperación de datos sea asíncrona.
5. Permitir al usuario añadir más datos personales dentro de su perfil (Institución, RRSS, ...)
6. Realizar una nueva página de usuarios con la nueva distribución.
7. Permitir al usuario ver sus propias estadísticas de uso.
8. Pantalla tipo *dashboard* con el estado en directo del sistema.

- **Marcas temporales**

  Este *sprint* se desarrolla entre el veintidós de febrero de dos mil veintidós y el uno de marzo de dos mil veintidós.

- ***Burndown chart***

  Con una duración de algo más de una semana, se han planificado un total de 58 puntos de historia para estos días. Las horas de trabajo han sido cercanas a las 45. La sensación del equipo de desarrollo después de haber finalizado el *sprint* es de un trabajo a ritmo constante finalizando tarea tras tarea, esta sensación se puede comprobar como en efecto ha sido así en la Figura A.12.

- ***Sprint review meeting***

  En este *sprint* no se han podido terminar todas las tareas, si bien en el servidor local en el que corre UBUMLaaS se ha podido desplegar Jenkins, el *pipeline* para que funcione correctamente no se ha podido terminar. Igual se buscan otras alternativas que además den soporte a elementos como las *badges* de GitHub y visualización de si pasan o no los tests en los propios *commits*.

  Las principales pantallas de administración van quedando mejor con cada *sprint*, más retoques se las van haciendo y el equipo de desarrollo

[2] Método de distribución de aplicaciones a los clientes con una cierta frecuencia mediante el uso de la automatización en las etapas del desarrollo de aplicaciones. Se trata de una solución para los problemas que se pueden generar en la integración del código nuevo en producción.

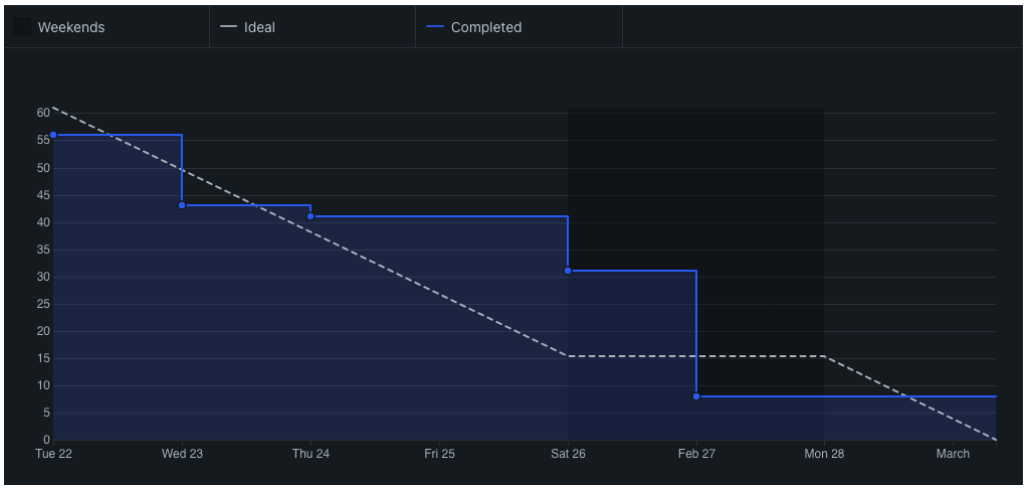

Figura A.12: *Burndown Chart Sprint 11.*

poco a poco comienza a sentirse más cómodo trabajando con lenguajes de marcas como es HTML, o de programación como JavaScript.

El número de horas invertidas en las que no se está programando como tal, sino que se requieren de aprendizaje previo a poder escribir código y hacer la tarea X que toque, sigue siendo alto en esta parte del proyecto.

Sprint 12: Name

- ***Planning meeting***

  Objetivos del duodécimo *sprint*:

  1. Se ha decidido dejar «en pausa» la traducción de UBUMLaaS a idiomas como el castellano o el francés. No se descarta retomarlo en un futuro o que sean líneas de trabajo futuro.
  2. Con la parte de administración ya más avanzada y con una cohesión mayor, se ha tomado la decisión de dar un «lavado de cara» a toda la aplicación, esto implica rehacer **todas** las páginas del *frontend* con el fin de que se adapten a la nueva guía de estilo de la aplicación.
  3. Realizar pequeños ajustes a ejes de gráficos.

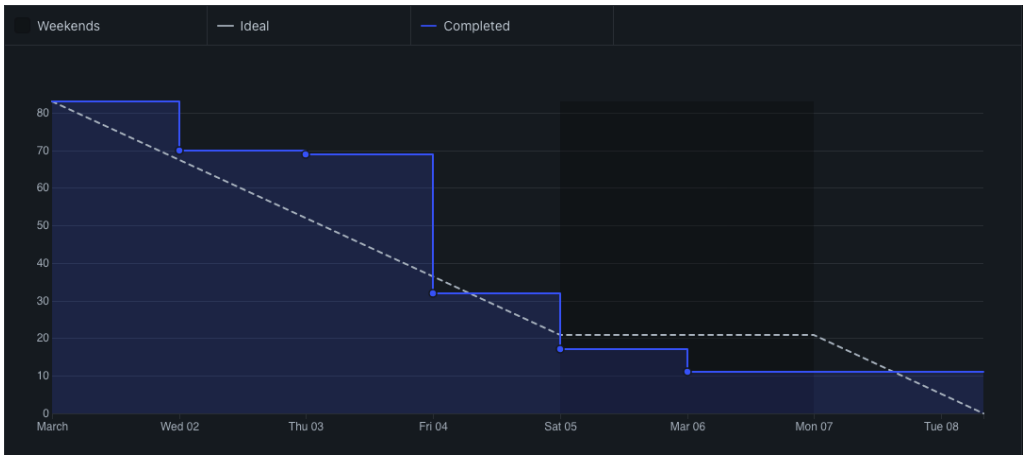

Figura A.13: *Burndown Chart Sprint 12.*

4. Decidir e implementar una forma de toma de datos en tiempo real del sistema anfitrión para en posteriores *sprints* visualizar esa información.
5. Implementación del algoritmo de aprendizaje Semi-Supervisado basado en picos de densidad, ver [15]

- **Marcas temporales**

  Este *sprint* se desarrolla entre el uno de marzo de dos mil veintidós y el ocho de marzo de dos mil veintidós.

- ***Burndown chart***

  El trabajo realizado en este *sprint* tal y como en la Figura A.13 se aprecia, ha sido superior a los anteriores, con un total de 83 puntos de historia y cerca de 45 horas invertidas. En esta ocasión el trabajo ha vuelto ha ser planificado «por lo alto» debido a la suposición de complejidad de implementación del algoritmo de aprendizaje Semi-Supervisado.

- ***Sprint review meeting*** En este *sprint* el equipo de desarrollo ha tenido la sensación que no «llegaba» a todo lo planificado, las reuniones llegan a un punto en el cuál se comenta trabajo, queda apuntado, y se intenta meter todo en tiempo y forma. Generando un cierto agobio en algunas situaciones que han impedido continuar con el trabajo al ritmo deseado.

En líneas generales se puede afirmar que el *frontend* ha sido rehecho entero, se han reutilizado formatos o formularios existentes por facilidad de uso a todos aquellos usuarios que ya la conocieran, pero a nivel de código prácticamente es nueva. Con un estilo mucho más moderno, fino y elegante.

La integración de CI/CD finalmente ha quedado hecha con elementos *cloud*, entre ellos se encuentran Travis-CI, Codebeat, SonarCloud y Codacy. Debido a que se ha rehecho toda la interfaz web de la aplicación, los tests existentes no pasan, es por ello que se tendrán que rehacer poco a poco, aunque no es uno de los elementos de mayor prioridad por el momento.

Sprint 13: Name

- ***Planning meeting***

  Objetivos del decimotercero *sprint*:

  1. Implementación del algoritmo de aprendizaje Semi-Supervisado basado en picos de densidad con filtrado, ver [9]
  2. Mejora inicial de la calidad del código.
  3. Panel *dashboard* de visualización de estado del sistema en tiempo real.
  4. Dar soporte a que el usuario pueda cambiar su foto de perfil.
  5. Realizar algunas pruebas de estrés para detectar puntos de rotura de la interfaz.
  6. Comenzar a escribir los Requisitos.
  7. Comenzar a escribir dentro del Diseño, el diagrama de casos de uso.
  8. Añadir a los aspectos relevantes los métodos que se están haciendo a sí como los cambios en la interfaz.
  9. Revisar comentarios hechos por Alvar en la memoria.

- **Marcas temporales**

  Este *sprint* se desarrolla entre el ocho de marzo de dos mil veintidós y el quince de marzo de dos mil veintidós.

- ***Burndown chart***

- ***Sprint review meeting***

***Sprint* n: Name**

- ***Planning meeting*** Objetivos del n *sprint*:
  1. Primero
  2. Segundo
- **Marcas temporales**
- ***Burndown chart***
- ***Sprint review meeting***

A.3. Estudio de viabilidad

Viabilidad económica

Viabilidad legal

Apéndice B

Especificación de Requisitos

B.1. Introducción

B.2. Objetivos generales

B.3. Catalogo de requisitos

B.4. Especificación de requisitos

Apéndice C

Especificación de diseño

C.1. Introducción

C.2. Diseño de datos

C.3. Diseño procedimental

C.4. Diseño arquitectónico

Apéndice D

Documentación técnica de programación

D.1. Introducción

D.2. Estructura de directorios

D.3. Manual del programador

D.4. Compilación, instalación y ejecución del proyecto

D.5. Pruebas del sistema

Apéndice E

Documentación de usuario

E.1. Introducción

E.2. Requisitos de usuarios

E.3. Instalación

E.4. Manual del usuario

Bibliografía

[1] Ricardo Barandela, Francesc J Ferri, and J Salvador Sánchez. Decision boundary preserving prototype selection for nearest neighbor classification. *International Journal of Pattern Recognition and Artificial Intelligence*, 19(06):787–806, 2005.

[2] Avrim Blum and Tom Mitchell. Combining labeled and unlabeled data with co-training. In *Proceedings of the eleventh annual conference on Computational learning theory*, pages 92–100, 1998.

[3] Henry Brighton and Chris Mellish. Advances in instance selection for instance-based learning algorithms. *Data mining and knowledge discovery*, 6(2):153–172, 2002.

[4] H Frank Cervone. Understanding agile project management methods using scrum. *OCLC Systems & Services: International digital library perspectives*, 2011.

[5] Geoffrey Gates. The reduced nearest neighbor rule (corresp.). *IEEE transactions on information theory*, 18(3):431–433, 1972.

[6] Marek Grochowski and Norbert Jankowski. Comparison of instance selection algorithms ii. results and comments. In *International Conference on Artificial Intelligence and Soft Computing*, pages 580–585. Springer, 2004.

[7] Peter Hart. The condensed nearest neighbor rule (corresp.). *IEEE transactions on information theory*, 14(3):515–516, 1968.

[8] Norbert Jankowski and Marek Grochowski. Comparison of instances seletion algorithms i. algorithms survey. In *International conference*

on artificial intelligence and soft computing, pages 598–603. Springer, 2004.

[9] Junnan Li, Qingsheng Zhu, and Quanwang Wu. A self-training method based on density peaks and an extended parameter-free local noise filter for k nearest neighbor. *Knowledge-Based Systems*, 184:104895, 2019.

[10] Huan Liu and Hiroshi Motoda. On issues of instance selection. *Data Mining and Knowledge Discovery*, 6(2):115, 2002.

[11] Dan Radigan. El backlog del producto: la lista de tareas pendientes definitiva, 2021.

[12] Julio Roche. Scrum: roles y responsabilidades, 2020.

[13] D Randall Wilson and Tony R Martinez. Reduction techniques for instance-based learning algorithms. *Machine learning*, 38(3):257–286, 2000.

[14] Di Wu, Mingsheng Shang, Xin Luo, Ji Xu, Huyong Yan, Weihui Deng, and Guoyin Wang. Self-training semi-supervised classification based on density peaks of data. *Neurocomputing*, 275:180–191, 2018.

[15] Di Wu, Mingsheng Shang, Xin Luo, Ji Xu, Huyong Yan, Weihui Deng, and Guoyin Wang. Self-training semi-supervised classification based on density peaks of data. *Neurocomputing*, 275:180–191, 2018.

[16] Yan Zhou and Sally Goldman. Democratic co-learning. In *16th IEEE International Conference on Tools with Artificial Intelligence*, pages 594–602. IEEE, 2004.

[17] Zhi-Hua Zhou and Ming Li. Tri-training: Exploiting unlabeled data using three classifiers. *IEEE Transactions on knowledge and Data Engineering*, 17(11):1529–1541, 2005.