Universidad Nacional de Misiones

Facultad de Ciencias Exactas, Químicas y Naturales

Trabajo Final de Maestría en Tecnologías de la Información

Selección de Técnicas Ágiles para la Gestión de Proyectos de Ciencia de Datos en Pequeñas y Medianas Organizaciones

Autor: **Martín Gustavo Rey**

Director: **Dr. Horacio Kuna**
Co-Director: **Dr. Emanuel Irrazabal**

AÑO 2021

Agradecimientos

A Rocío, mi compañera de vida por todo su apoyo, acompañamiento y, principalmente, paciencia en este camino.

Al Dr. Horacio Kuna, por la renovada confianza, su guía constante a lo largo de estos años y la motivación para alcanzar este objetivo.

Al Dr. Emanuel Irrazabal, por la confianza depositada en esta idea, sus aportes en todo el proceso y el apoyo constante.

A todos los involucrados en hacer posible esta maestría por brindarnos esta oportunidad de crecimiento académico, profesional y personal.

A mis compañeros de cursado, parte importante del camino recorrido y por recorrer.

A mis compañeros de trabajo y autoridades de la Facultad de Ciencias Económicas por posibilitar la realización de la validación de la propuesta presentada.

Resumen

En este trabajo se presenta un marco de trabajo para la gestión de proyectos de ciencia de datos basado en métodos ágiles. El mismo se elabora teniendo en cuenta las particularidades de este tipo de proyectos y seleccionando tanto las tareas de gestión como las herramientas tecnológicas de soporte aplicables. La gestión de proyectos de minería y/o análisis de datos puede ser realizada de diferentes maneras, a lo largo del tiempo, se han generado diversas metodologías o modelos de procesos para tales objetivos. Sin embargo, el marco de trabajo de mayor utilización en la historia reciente, CRISP-DM, es uno que presenta dificultades en este aspecto y se centra en cuestiones de índole técnico. Las interacciones entre agilidad y ciencia de datos presentan en la actualidad una alternativa al enfoque clásico de gestión, sin embargo no ofrecen una selección de herramientas que brinden soporte para las tareas de administración a realizar. Además de presentar limitaciones en su adopción en pequeñas y medianas organizaciones. El marco de trabajo presentado integra aspectos de gestión ágil junto a la propuesta de herramientas software para brindar soporte en su aplicación y provee elementos de adaptación para su utilización en organizaciones como las mencionadas anteriormente. El mismo ha sido validado mediante su aplicación en un proyecto real y a través de un marco comparativo para metodologías de ciencia de datos. Como resultado se ha podido comprobar la utilidad y validez del marco de trabajo propuesto y la integración del mismo con las herramientas de soporte seleccionadas.

Palabras clave: ciencia de datos, modelos de gestión, agilidad.

Abstract

In this work, a data science project management framework based on agile methods is presented. It is prepared taking into account the characteristics of this kind of projects along with the management tasks and the applicable support tools. The management of data mining and/or data analysis projects can be carried out in different ways, over time, various methodologies or process models have been generated for such objectives. However, the most widely used framework in recent history, CRISP-DM, is one that is challenging in this regard and focuses on technical issues. The interactions between agility and data science currently present an alternative to the classic management approach, however they do not offer a selection of tools that provide support for the administration tasks to be performed. In addition to present limitations in its adoption in small and medium organizations. The presented framework integrates agile management aspects together with the proposal of software tools to provide support in its application and provides adaptable elements for use in organizations such as those mentioned above. It has been validated through its application in a real project and through a comparative framework for data science methodologies. As a result, it has been possible to verify the usefulness and validity of the proposed framework and its integration with the selected support tools.

Key words: data science, management models, agility.

Índice General

Índice de Figuras **VII**

Índice de Tablas **IX**

1 Introducción **9**

1.1 Introducción 9

1.2 Contexto 10

1.3 Estructura del documento 11

2 Marco teórico **13**

2.1 Ciencia de datos 13

2.1.1 Evolución del concepto 13

2.2 Métodos para la gestión de proyectos de ciencia de datos 15

2.2.1 Descubrimiento de Conocimiento en Bases de Datos 15

2.2.2 CRISP-DM: Cross-Industry Standard Process for Data Mining . . . 17

2.2.3 Metodologías SEMMA y Catalyst 20

2.2.3.1 SEMMA: Sample, Explore, Modify, Model and Assess . . 20

2.2.3.2 Catalyst 21
2.3 Métodos ágiles para el desarrollo de software 22
2.3.1 El manifiesto ágil 22
2.3.2 Principales metodologías ágiles 24
2.3.2.1 Scrum 24
2.3.2.2 XP: eXtreme Programming 25
2.3.2.3 Lean Software Development 27
2.3.2.4 Test Driven Development 28
2.3.2.5 Kanban 29
2.3.2.6 ScrumBan = Scrum + Kanban 30
2.3.2.7 Heart of Agile 31
2.3.2.8 3x: Explore, Expand and Extract 31
2.3.2.9 Modern Agile 32
2.4 Métodos ágiles para proyectos de ciencia de datos 34
2.4.1 Prácticas ágiles en proyectos de ciencia de datos 34
2.4.2 ASUM-DM: Analytics Solutions Unified Method for Data Mining . 37
2.4.3 TDSP: Team Data Science Process 38
2.4.4 Scrum y ciencia de datos 40

3 Descripción del problema **43**

3.1 Generalidades 43
3.2 Objetivos 44
3.3 Alcance y limitaciones 45

4 Solución propuesta **47**

4.1 Introducción 47

4.2 Relevamiento de técnicas ágiles aplicables a proyectos de ciencia de datos 49

4.3 Relevamiento de enfoques de gestión de proyectos de ciencia de datos que aplican agilidad 53

4.3.1 Puntos en común entre las diferentes estrategias de integración de agilidad en ciencia de datos 58

4.4 Elaboración de la propuesta de solución 62

4.4.1 Pequeñas y medianas organizaciones 62

4.4.2 Definición de la propuesta de gestión ágil para proyectos de ciencia de datos 65

4.4.2.1 Inicio del proyecto 68

4.4.2.2 Iteración cero 70

4.4.2.3 Inicio de una iteración 73

4.4.2.4 Desarrollo de una iteración 75

4.4.2.5 Cierre de una iteración 79

4.4.2.6 Despliegue 82

4.4.2.7 Situaciones a considerar 84

4.5 Herramientas software de soporte a la propuesta 86

4.5.1 Para la gestión del proyecto 87

4.5.2 Para la gestión del versionado del producto 89

4.5.3 Para la gestión de la automatización y los entornos de trabajo . . . 90

4.5.4 Resumen 93

5 Validación **95**

5.1 Introducción 95

5.2 Caso de estudio 97

5.2.1 Inicio del proyecto 98

5.2.1.1 Generación del backlog del producto 100

5.2.1.2 Planificación de versiones 103

5.2.1.3 Organización del trabajo 105

5.2.2 Iteración cero 105

5.2.3 Iteración uno 109

5.2.3.1 Inicio de la iteración 109

5.2.3.2 Desarrollo de la iteración 112

5.2.3.3 Cierre de la iteración 115

5.2.4 Iteración dos 116

5.2.4.1 Inicio de la iteración 116

5.2.4.2 Desarrollo de la iteración 119

5.2.4.3 Cierre de la iteración 123

5.2.5 Análisis de resultados 124

5.3 Evaluación comparativa de la propuesta 126

5.3.1 Comparación realizada 127

5.3.2 Análisis de los resultados 136

5.4 Resultados generales de la validación 137

6 Conclusiones **139**

6.1 Conclusiones del trabajo 139

6.2 Futuras lineas de acción 141

Bibliografía **143**

Índice de Figuras

2.1 Fases del proceso de KDD. 17

2.2 Fases del modelo de procesos CRISP-DM. 20

4.1 Fases de la propuesta desarrollada. 68

4.2 Detalle de la iteración cero. 72

4.3 Tareas de inicio de cada iteración. 75

4.4 Tareas ejecutables en el desarrollo de cada iteración. 79

4.5 Tareas de cierre de cada iteración. 80

5.1 Estructura de directorios del proyecto. 107

5.2 Arquitectura inicial del proyecto. 108

5.3 Historia de usuario del proyecto generada como épica en Jira. 109

5.4 Items de *backlog* vinculados a una épica en Jira. 111

5.5 Tareas vinculadas a un item de *backlog* en Jira. 112

5.6 Vista del tablero al final de la primera iteración. 115

5.7 Items de *backlog* para la segunda iteración. 118

5.8 Tareas de un item de *backlog* para la segunda iteración. 119

5.9 Hoja de ruta al final de la segunda iteración. 122

5.10 Vista de la PoC del proyecto en funcionamiento. 122

Índice de Tablas

4.1 Prácticas de gestión y su relación con métodos ágiles. 54

4.2 Prácticas técnicas y su relación con métodos ágiles. 55

4.3 Características de PyMEs según AFIP. 63

4.4 Características de PyMEs según el OPSSI de la CESSI. 63

4.5 Fase de inicio del proyecto. 69

5.1 Evaluación comparativa: Nivel de detalle en la descripción de las actividades. 128

5.2 Evaluación comparativa: Escenarios de aplicación. 129

5.3 Evaluación comparativa: Actividades específicas que componen cada fase. Análisis del problema. 129

5.4 Evaluación comparativa: Actividades específicas que componen cada fase. Selección y preparación de los datos. 130

5.5 Evaluación comparativa: Actividades específicas que componen cada fase. Modelado. 130

5.6 Evaluación comparativa: Actividades específicas que componen cada fase. Evaluación. 131

5.7 Evaluación comparativa: Actividades específicas que componen cada fase. Implementación. 131

5.8 Evaluación comparativa: Actividades específicas que componen cada fase. Resumen. 132

5.9 Evaluación comparativa: Actividades de dirección del proyecto. Gestión del alcance. 132

5.10 Evaluación comparativa: Actividades de dirección del proyecto. Gestión del tiempo 133

5.11 Evaluación comparativa: Actividades de dirección del proyecto. Gestión del costo 133

5.12 Evaluación comparativa: Actividades de dirección del proyecto. Gestión del equipo de trabajo 134

5.13 Evaluación comparativa: Actividades de dirección del proyecto. Gestión del riesgo 134

5.14 Evaluación comparativa: Actividades de dirección del proyecto. Resumen 135

5.15 Evaluación comparativa: Resultados generales. 135

Capítulo 1

Introducción

En este capítulo se explica la motivación que llevó al desarrollo del presente trabajo junto a una descripción general del contexto de su ejecución y de la estructura del documento.

1.1 Introducción

Es posible definir a la ciencia de datos como un punto de encuentro entre diferentes disciplinas y actividades que tienen por objetivo el análisis y la utilización de datos como soporte para la toma de decisiones.

En la historia reciente los volúmenes de datos, la complejidad de las técnicas aplicables y la criticidad de los resultados a obtener en iniciativas de explotación de información ha ido en aumento. Esto llevó al desarrollo de diferentes metodologías o marcos de procesos que permitieran realizar un seguimiento adecuado de la ejecución de un proyecto de este tipo. Sin embargo, en la actualidad las más utilizadas continúan siendo las de principios de siglo que presentan algunas dificultades en lo que respecta a la gestión de los proyectos, al estar más orientadas a cuestiones técnicas.

En paralelo, la industria del desarrollo de software fue creando nuevos métodos para gestión de proyectos que pasaron a denominarse ágiles. Estos se basaron en

criterios de éxito y estrategias de gestión diferentes a los esquemas predictivos históricamente aplicados. Y han demostrado ser igualmente efectivos e ir ganando volumen de uso en el mercado.

Es en ese contexto donde se empezaron a ver interacciones entre la gestión ágil de proyectos y su aplicación a iniciativas de ciencia de datos. Sin embargo, no se observa en las mismas un enfoque completo al incluir herramientas que brinden soporte a las técnicas empleadas o lo hacen sobre plataformas que no son totalmente abiertas. Además, se identifica otro inconveniente a la hora de adaptar las propuestas encontradas a organizaciones o proyectos de menor tamaño, en donde se presentan realidades diferentes a las grandes empresas.

De esta manera, el presente trabajo final de maestría busca seleccionar diferentes técnicas aplicables a la gestión de proyectos de ciencia de datos. En primer lugar, para lograr obtener los beneficios planteados por este tipo de estrategias en la industria del software, tales como la visibilidad y trazabilidad de los avances del proceso, la entrega constante de resultados de valor al cliente y la adaptación a cambios en el contexto. Pero también para integrar en la selección a un conjunto de herramientas software que brinden soporte en su utilización cotidiana. Todo esto enfocado a su aplicación y, con capacidad de adaptación, a los entornos de trabajo de pequeñas y medianas organizaciones.

1.2 Contexto

El trabajo documentado aquí ha sido financiado parcialmente por una beca del Programa Estratégico de Formación de Recursos Humanos en Investigación y Desarrollo (PERHID) otorgada por el Consejo Universitario Nacional (CIN).

El tema en cuestión se encuentra dentro de las líneas de investigación del Instituto de Investigación, Desarrollo e Innovación en Informática (IIDII) de la Facultad de Ciencias Exactas, Químicas y Naturales (FCEQyN) de la Universidad Nacional de Misiones.

Asímismo, el caso de estudio presentado en el apartado de validación fue desarrollado en el marco de la Dirección de Tecnologías para la Gestión de la Facultad de

Ciencias Económicas de la misma Universidad, contando con las autorizaciones correspondientes para tal tarea.

1.3 Estructura del documento

El resto del documento se organiza de la siguiente manera:

- En el capítulo dos se presentan las bases teóricas consultadas para el desarrollo de la propuesta. Se describen los grandes temas involucrados: ciencia de datos y sus metodologías de gestión, los métodos ágiles aplicables para el desarrollo de software y las iniciativas de interacción entre ambos mundos.
- En el capítulo tres se describe el problema a resolver mediante el desarrollo del presente trabajo. Se exponen los objetivos, el alcance y limitaciones del mismo.
- En el capítulo cuatro se comenta el camino recorrido para la generación del marco de trabajo resultante. Se presentan las selecciones progresivas de técnicas y métodos tanto de gestión ágil como de ciencia de datos en sí. Se detallan todos los componentes de la propuesta, su interacción y las herramientas aplicables para brindar soporte a la misma.
- En el capítulo cinco se desarrolla la validación de la propuesta mediante enfoques correspondientes al tipo de producto generado.
- En el capítulo seis se exponen las conclusiones a las que se ha llegado con el final del trabajo y los posibles caminos a seguir como futuras líneas de investigación.

Capítulo 2

Marco teórico

En este capítulo se presentan las bases teóricas del trabajo. Se inicia por establecer una convención de referencia para ciencia de datos y otras definiciones aplicables para el resto del documento. En particular, se comienza por una reseña de los diferentes modelos para gestión de proyectos de ciencia de datos y se continua con los modelos de gestión ágil de proyectos de desarrollo de software. Finalmente se presentan las iniciativas actuales para la unificación de ambas áreas, los proyectos de ciencia de datos gestionados a partir de la utilización de prácticas ágiles.

2.1 Ciencia de datos

En la presente sección se trabaja sobre la definición de *"ciencia de datos"*, dado que existen diferentes acepciones con mayor o menor grado de aceptación se pretende unificar, a fines prácticos, un concepto general para su uso a lo largo del documento.

2.1.1 Evolución del concepto

Existen diferentes definiciones para el término o vocablo *"ciencia de datos"*, cada una de ellas con voces a favor y en contra [1, 2]. La más simplista de ellas sería la que marca que *"la ciencia de datos es el estudio de los datos"*[1]. Sin embargo, a fines prácticos,

en el presente trabajo se utilizará la siguiente: *"es un nuevo campo interdisciplinario que unifica estadísticas, informática, comunicaciones, administración y sociología para estudiar los datos y su entorno a fin de transformar datos en ideas y colaborar en la toma de decisiones siguiendo una metodología o línea de pensamiento definida por la siguiente progresión datos-a-conocimiento-a-sabiduría"* [1]. De esta manera se puede establecer la relación::

ciencia de datos = estadística + informática + comunicaciones + manejo-de-datos

Para revisar la evolución de los conceptos involucrados se puede llegar a retroceder hasta los años 60 / 70 del siglo XX cuando se comenzó a considerar al análisis de datos con una disciplina científica. Relacionada inicialmente con los métodos estadísticos y la matemática, pero con la intención de convertir a los datos en información y conocimiento [3, 4]. Desde ese momento se relaciona a la ciencia de datos con el procesamiento de la información y el análisis exploratorio de datos [1, 5].

En los años 80 / 90 se instala el concepto de minería de datos y descubrimiento de conocimiento, como procesos para obtener patrones previamente desconocidos y de potencial utilidad a partir de datos [6, 7]. En tales años, se produce un incremento en el interés sobre el tema por parte del ámbito empresarial, principalmente respecto a su aplicación con fines de marketing [8]. Se reconoce que diversas compañías generan y almacenan grandes cantidades de datos a fin de analizarlos para realizar predicciones y utilizar tal conocimiento para potenciar sus ventas. El análisis de datos se presenta como una ciencia multidisciplinaria en la que se analizan datos en forma cuantitativa y cualitativa para generar nuevas conclusiones sobre ellos aplicando métodos descriptivos o predictivos [1].

Con el inicio del nuevo milenio el uso de datos para obtener ventajas competitivas a nivel de empresas se vuelve más frecuente [9]. Con la consecuente generación de perfiles para los denominados *"científicos de datos"*, especificando algunas de las habilidades necesarias para este tipo de trabajo [10, 11, 12, 13]. Entre todas las habilidades que componen a un científico de datos se pueden mencionar:

- Conocimientos de matemática y estadística

- Conocimientos de bases de datos
- Conocimientos de programación
- Conocimientos de algoritmos de análisis y visualización de datos
- Capacidad para identificar patrones en datos

2.2 Métodos para la gestión de proyectos de ciencia de datos

La aplicación de diversos métodos para obtener información o conocimiento a partir de un conjunto de datos generó la necesidad de un marco de trabajo común. El objetivo detrás de su aplicación fue la obtención de resultados reproducibles, verificables y con ciertos niveles de calidad [6]. En su definición se incluyen las etapas a seguir en un proyecto de este tipo, las prácticas de cada una de ellas, sus entradas y salidas.

2.2.1 Descubrimiento de Conocimiento en Bases de Datos

El proceso de KDD, sigla para *Knowledge Discovery in Databases*, fue una primera aproximación a un modelo de gestión de proyectos de explotación de información. Se conforma de nueve etapas que abarcan desde la comprensión del dominio de trabajo hasta la presentación de los resultados. Las fases del modelo se describen a continuación [6]:

1. **Entendimiento del dominio**: se comienza por intentar comprender los objetivos del proceso de KDD desde la perspectiva del cliente o principal consumidor de los resultados que se prevé obtener.
2. **Creación del conjunto de datos**: se debe obtener o construir el *dataset* que será analizado a través del proceso completo. Usualmente podría implicar la unificación de diferentes fuentes y la selección de los datos potencialmente adecuados para cumplir los objetivos definidos.

3. **Limpieza y preprocesado de los datos**: en este punto, las actividades se centran en la identificación de datos erróneos, faltantes y/o inconsistentes que puedan dificultar los pasos siguientes. Una vez detectados, se determinan estrategias a seguir para su tratamiento, pudiendo incluir: eliminación, establecimiento de valores según criterios estadísticos, inyección de valores desde fuentes externas, entre otras.

4. **Transformación de los datos**: se aplican reducciones dimensionales (columnas y/o filas) sobre el *dataset* generado previamente a fin de obtener la mejor representación de los datos para los objetivos planteados.

5. **Selección de métodos de minería de datos**: se seleccionan las técnicas de minería de datos a emplear en el procesamiento del *dataset*. En este paso se hace referencia al problema de explotación de información a resolver en forma genérica: clasificación, regresión, predicción, descubrimiento de grupos, entre otros [14].

6. **Determinar técnicas de minería de datos**: con el tipo de problema identificado se procede con la selección de los algoritmos a aplicar para la extracción de patrones. También se establecen, al menos inicialmente, los valores de los parámetros de cada técnica empleada.

7. **Aplicar técnicas de minería de datos**: concretamente, este paso se refiere a la ejecución de las técnicas del paso anterior sobre el conjunto de datos, recopilando los resultados que se obtengan para su posterior análisis.

8. **Interpretar el conocimiento obtenido**: con asistencia de un experto en el dominio del problema se comienza con el análisis de los resultados obtenidos. Se busca determinar la utilidad de los patrones presentes en los datos para los objetivos planteados.

9. **Utilizar el conocimiento obtenido**: en esta última fase del proceso se determina de qué manera va a ser empleado el conocimiento obtenido y se ejecutan las acciones necesarias para ello. Entre las opciones posibles se pueden mencionar: integración en un sistema de soporte a la toma de decisiones, usarlo como entrada para otro proceso de similares características, documentar los patrones para su comparación con otros obtenidos previamente, entre otras.

Las fases del proceso de KDD no necesariamente siguen una progresión lineal (ver figura 2.1), sino que pueden existir bucles e iteraciones entre etapas a fin de refinar cuestiones relativas al proceso, a los datos o a los resultados.

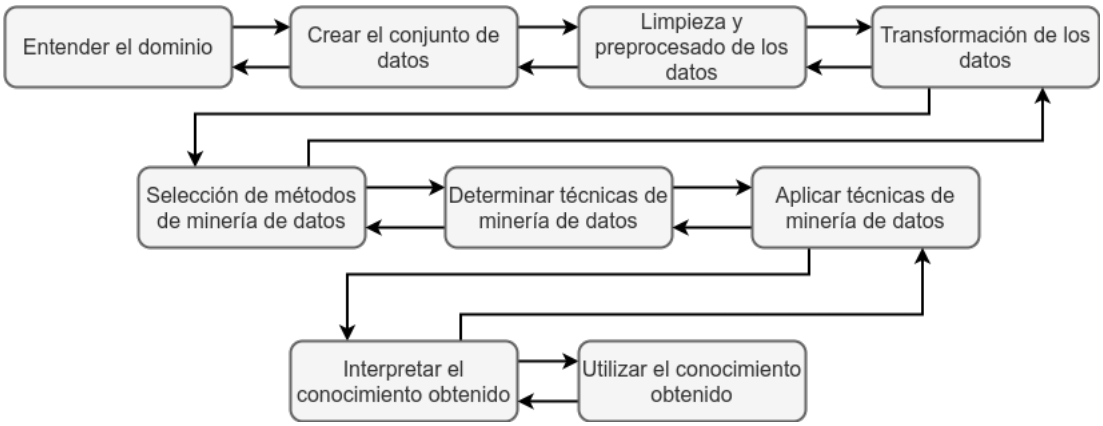

Figura 2.1: Fases del proceso de KDD.
Fuente: Elaboración propia basada en [6]

2.2.2 CRISP-DM: Cross-Industry Standard Process for Data Mining

CRISP-DM es un modelo de procesos para la gestión de proyectos de minería de datos [15], es considerado el estándar de la industria [16, 17]. Su principal diferencia con otros modelos de procesos como SEMMA [14] y Catalyst (más conocida como P3TQ) [18] radica en la independencia de una herramienta software que brinde soporte a sus prácticas [19]. En términos generales, su estructura se compone de etapas por las que debe pasar un proyecto de minería de datos, tareas a ejecutar en cada una de las fases y las conexiones que se definen entre las actividades planteadas para constituir el flujo de resultados del proceso [15]. En su definición, CRISP-DM se divide en dos componentes, por un lado un modelo de referencia que resume las fases, los diferentes niveles de tareas y los resultados esperados para el proyecto. Por otra parte, cuenta con una guía de usuario que establece recomendaciones para la ejecución de las actividades previstas dentro de un proyecto en diversos contextos [17]. El modelo de referencia estructura los componentes de la metodología en forma jerárquica, estableciendo cuatro niveles para la definición de las tareas disminuyendo progresivamente el grado de abstracción:

- Fases

- Tareas genéricas
- Tareas específicas
- Instancias de procesos

En los primeros dos niveles se encuentran los elementos que proveen mayor flexibilidad al modelo. Esto permite adaptarlo a diferentes escenarios sin que esto implique una pérdida de completitud en general para el proceso de explotación de información. Los siguientes niveles constituyen la descripción de acciones concretas a ejecutar en función de las tareas genéricas junto a opciones para su ejecución y los lineamientos para el registro de sus resultados en el marco del proyecto en ejecución.

Las fases, correspondientes al nivel más alto de abstracción de CRISP-DM, se describen a continuación y se presentan en la figura 2.2 [15]:

1. **Comprensión del negocio**: en esta fase se busca conocer los objetivos del proyecto desde una perspectiva de negocios, se define el problema de minería de datos a resolver junto a una primera versión de la planificación del mismo. En esta fase se establecen los criterios de éxito del proyecto, sobre los cuales serán evaluados posteriormente sus resultados, los recursos disponibles para su ejecución y una evaluación preliminar de costos y beneficios.
2. **Comprensión de los datos**: se inicia con la recolección de los datos y se analizan los mismos a fin de comprender sus características. A partir de estas acciones se podrán identificar problemas de calidad, particiones sobre las que sea de particular interés ejecutar algún método de análisis, entre otras cuestiones.
3. **Preparación de los datos**: una vez que se cuenta con el conjunto general de datos a procesar, se busca construir el *dataset* objetivo sobre el cual se aplicarán las técnicas de minería de datos. Las operaciones a aplicar para cumplir este objetivo pueden incluir: selección de columnas y/o filas, transformación de valores, generación de atributos, limpieza de datos erróneos, unificación de fuentes de datos, entre otras.
4. **Modelado**: sobre el *dataset* con las adaptaciones necesarias aplicadas se prosigue con la aplicación de una o más técnicas de explotación de información a fin de obtener patrones o modelos aplicables al problema en cuestión. La cantidad y tipo

de técnicas seleccionadas serán definidas en función de los requerimientos del caso. Para cada una de ellas se deberán establecer los umbrales para los valores de sus parámetros y, posiblemente, realizar algunas iteraciones de prueba para su óptima calibración. En caso de ser necesario, se podrá volver a ejecutar una o más actividades de las fases previas para lograr una mejora en la calidad de los resultados. Los modelos se evaluarán para determinar si se han alcanzado los criterios de éxito establecidos previamente.

5. **Evaluación**: la ejecución de las tareas de la fase anterior generará una serie de resultados cuyo análisis determina su utilidad en relación al cumplimiento de los objetivos del proyecto. La evaluación abarcará también al proceso ejecutado para generar tales resultados, proponiendo modificaciones al mismo en próximas iteraciones si fuera oportuno. En esta etapa, se establecen los pasos a seguir, tanto en esta iniciativa como en otras que guarden relación con los objetivos de negocio planteados.

6. **Despliegue**: en la fase final del proceso se organizan los resultados obtenidos en forma de conocimiento, patrones o modelos para su presentación y posible integración dentro de algún otro producto de la organización. Las actividades de esta fase dependerán de la naturaleza del problema y los objetivos planteados inicialmente. Se incluirán actividades de seguimiento y monitoreo del uso o explotación de los resultados para su uso en posteriores iniciativas.

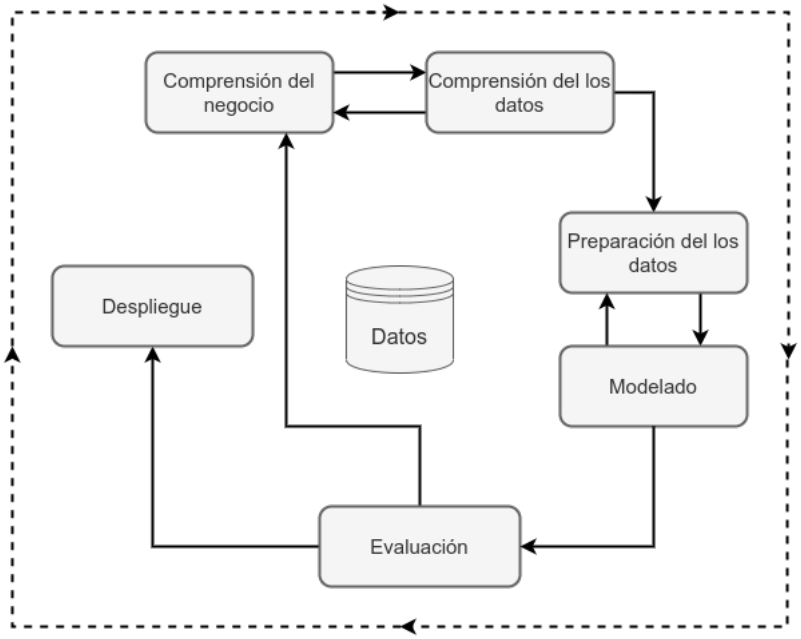

Figura 2.2: Fases del modelo de procesos CRISP-DM.
Fuente: Elaboración propia basada en [15]

2.2.3 Metodologías SEMMA y Catalyst

En ambos casos, se trata de metodologías aplicables a proyectos de minería de datos que, aunque contemporáneas a CRISP-DM, no han manifestado el mismo impacto en la comunidad y su uso ha sido más reducido [16, 17]. Los aspectos principales de ambas metodologías se describen a continuación:

2.2.3.1 SEMMA: Sample, Explore, Modify, Model and Assess

Es un proceso para descubrir patrones ocultos en grandes volúmenes de datos a través de la secuencia de operaciones de su nombre (muestreo, exploración, modificación, modelado y evaluación). Fue creada por SAS y su punto más criticable radica en la utilización de una herramienta software en forma exclusiva para su aplicación [16].

La metodología se centra en aspectos prácticos o técnicos, relegando a un segundo plano cuestiones relativas al análisis o comprensión del problema sobre el que se

este trabajando. Inicia en forma directa con la extracción de una muestra de los datos a analizar, determinando el nivel de confianza de la misma con respecto a su representación del problema en estudio. Con la muestra seleccionada se procede a su exploración, en un intento de identificar relaciones existentes entre valores a través de diferentes técnicas o herramientas de visualización de datos. Esta segunda fase finaliza cuando se obtiene el conjunto de variables objetivo para el análisis previsto que serán la entrada a la fase siguiente [14].

En la tercera fase, las actividades prevén la manipulación de los datos a fin de definir el formato que resulte más compatible con las técnicas a emplear a continuación. Con este *dataset* se inicia el análisis y modelado de los datos, las técnicas a emplear podrán variar en función de las características del proyecto, sin embargo el objetivo es generar un modelo que permita predecir el valor de los atributos objetivo con un nivel mínimo de certeza [14].

Finalmente, la última fase consiste en la evaluación de los resultados obtenidos en la etapa previa para determinar si los mismos pueden ser utilizados por los interesados en el proyecto [14].

2.2.3.2 Catalyst

Más conocida como P3TQ a partir de la sigla de *Product, Place, Price, Time and Quantity*, se descompone en dos modelos: uno orientado al negocio y otro orientado a la explotación de información. El primero de ellos se compone de una guía para el desarrollo y la construcción de uno o más modelos que describan el problema o la oportunidad de negocio que da origen al proyecto de explotación de información. El segundo, constituye un conjunto de pasos para ejecutar operaciones de extracción de conocimiento en función del contexto descripto previamente [18].

El modelado del negocio relacionado al proyecto se plantea en la metodología a partir del análisis del escenario sobre el cual comienza a trabajar. En este sentido, existen diversos puntos de partida como ser [14, 18]:

- Un conjunto de datos a explorar en busca de información nueva y de potencial utilidad.

- Una oportunidad de obtener una ventaja frente a la competencia o de resolver alguna problemática interna.
- Identificar procesos en la organización con posibilidad de mejorar su toma de decisiones con información.
- Entre otros.

El modelo de explotación de información se compone de actividades a ejecutar, tales como: preparación de los datos, selección inicial de herramientas y modelado preliminar, ejecución o aplicación de técnicas de explotación de información, evaluación de resultados y presentación de resultados. En cada una de estas acciones se listan pasos en un mayor nivel de detalle que sirven como guía para su ejecución, evaluación de avance, entre otros [16, 18].

2.3 Métodos ágiles para el desarrollo de software

2.3.1 El manifiesto ágil

En febrero de 2001, un conjunto de expertos en desarrollo de software se reunieron para establecer una serie de cambios en la actividad a fin de combatir las limitaciones de este tipo de proyectos que existían hasta el momento. Estas limitaciones, formaban parte de la denominada "*crisis del software*", entre las que se pueden mencionar: secuencialidad en las actividades, centralidad en la documentación, procesos rígidos y alta dependencia de un plan de actividades detallado al inicio de un proyecto de desarrollo, entre otras.

Como resultado de la reunión surgió el **Manifiesto Ágil para el Desarrollo de Software** [20], que se compuso de una serie de valores y prácticas para cambiar la forma en la que el software es desarrollado y los proyectos de este tipo son gestionados.

A continuación se transcribe parcialmente el documento [20]:

"*Estamos descubriendo formas mejores de desarrollar software tanto por nuestra propia experiencia como ayudando a terceros. A través de este trabajo hemos aprendido a valorar:*

Individuos e interacciones *sobre procesos y herramientas*

Software funcionando *sobre documentación extensiva*

Colaboración con el cliente *sobre negociación contractual*

Respuesta ante el cambio *sobre seguir un plan*

Esto es, aunque valoramos los elementos de la derecha, valoramos más los de la izquierda." [20]

A partir de lo establecido en el manifiesto, se puede identificar una serie de prácticas para guiar el desarrollo de software, entre las cuales se destacan [20]:

- Entrega de software en forma temprana y continua con el foco en la satisfacción del cliente.
- Los cambios son aceptados en todo momento del desarrollo ya que representan oportunidades para lograr una ventaja competitiva.
- Las entregas de software funcional se realizan con una frecuencia de entre dos a ocho semanas.
- Los representantes del negocio se integran al equipo de desarrollo para el trabajo día a día durante el proyecto.
- Los equipos de trabajo deben estar continuamente motivados, contando con el apoyo y las herramientas que generen un entorno de trabajo adecuado.
- El avance del proyecto se medirá en software que se encuentre funcionando.

Previo a la presentación del manifiesto, diversas metodologías o marcos de procesos[1] recomendaban algunos de los principios, valores y/o prácticas mencionadas, de igual manera, fueron adoptados como la base para las metodologías que se generaron posteriormente.

[1]A lo largo del presente documento, por una cuestión de practicidad ambos términos serán usados en forma indiferente dada su aceptación en la industria o comunidad del desarrollo de software.

2.3.2 Principales metodologías ágiles

En este apartado, se describen algunas de las metodologías ágiles de mayor trascendencia en la industria. Sobre el final del listado se incluyen referencias a algunos *frameworks* más recientes en el tiempo que podrían ser considerados.

2.3.2.1 Scrum

Planteado desde su definición como un método de gestión de proyectos, Scrum se presentó como una alternativa frente a los problemas que para la década de 1990 presentaba la gestión clásica de proyectos [21, 22]. Sus diferencias con los métodos predictivos o tradicionales se centraron en los siguientes aspectos [22, 23, 24]:

- Se trabaja con un único equipo que reúne todos los perfiles necesarios para el desarrollo del proyecto. Los diferentes puntos de vista sobre el problema a resolver se consideran una oportunidad para lograr un mejor producto.
- No se trabaja de forma secuencial, dividiendo el trabajo en fases, sino que las tareas son ejecutadas en el momento en que son necesarias, definiendo iteraciones. No se abarca el total del alcance en una única vez sino que se realiza en fracciones en función de la necesidad.
- Los requisitos del proyecto serán definidos con mayor detalle a medida que se avanza en el proyecto y se tiene más conocimiento sobre las características del problema a resolver. Inclusive, se acepta incorporar nuevos requerimientos en etapas avanzadas.
- La definición completa del proyecto no es conocida desde su inicio sino que se descubre a través de las iteraciones de su ejecución.

En Scrum, la incertidumbre con respecto a no contar con un plan detallado para la ejecución de las actividades del proyecto y la capacidad del equipo de trabajo de organizarse en forma autónoma, son altamente valoradas. El enfoque de trabajo basado en las instancias de control de avances con la participación del cliente, requiere de un conjunto de acciones de comunicación, difusión del conocimiento y evaluación

que permitan gestionar el proyecto a todos los interesados en el mismo. Este último aspecto se complementa con la transformación de las fases tradicionales de un proyecto de desarrollo en actividades dentro de una iteración que pueden ser utilizadas según lo crea conveniente el equipo de trabajo [22, 23, 24, 25].

En relación a lo establecido previamente, Scrum no abarca temas como son: pruebas, diseño, arquitectura y otras cuestiones específicas del desarrollo de software sino que deriva esas prácticas a otras metodologías como las que se presentarán más adelante. Más allá de esto, su aplicación en este tipo de proyectos se justifica a partir de que su ciclo de vida iterativo e incremental, los eventos que define, los roles a cumplir por el equipo y los artefactos que se generan se adecuan para obtener resultados y adaptarse a cambios en el contexto a medida que se desarrolla el producto [26, 27, 28].

Los **eventos** que plantea la metodología se vinculan a los *sprints* (las iteraciones) y una serie de reuniones relacionadas a los mismos: de planificación, diaria, de revisión y de retrospectiva. Constituyen las instancias de colaboración de todos los involucrados en el proyecto. Por otra parte, los **artefactos** que se generan en la ejecución del proyecto: *product backlog*, *sprint backlog*, los incrementos en el desarrollo del producto y un concepto asociado que es la definición de terminado o hecho[2]. Mientras que los **roles** que se asignan a los involucrados en el proyecto pueden ser: el equipo de Scrum, el dueño del producto, el equipo de desarrollo, el Scrum *master* y los *stakeholders* o clientes del proyecto [22, 26]. La conjunción de estos tres elementos en la ejecución de la metodología permite obtener los beneficios mencionados previamente y llevar a cabo una gestión ágil de un proyecto, tanto en general como de desarrollo de software [27].

2.3.2.2 XP: eXtreme Programming

Definida como "*un proceso ligero, de bajo riesgo, flexible, predecible, científico y divertido de desarrollar software*" [29, 30], se trata de una metodología diseñada para equipos de tamaño reducido (alrededor de diez integrantes) que tiene como una de sus mayores características la flexibilidad para adaptarse a los cambios. Plantea un

[2]Se trata de una definición acordada por un equipo de trabajo que representa que un item a desarrollar se encuentra finalizado. Esta definición, según el equipo y su contexto, podría variar por lo que no es estricta en cuanto a qué incluye.

esquema de trabajo iterativo e incremental donde se prioriza el desarrollo del producto (código), y los *tests* son parte del mismo al punto que se desarrollan en paralelo y no posteriormente. También se formaliza el uso de historias de usuario[3] como instrumento de comunicación entre el cliente y el equipo de desarrollo [29, 30].

La metodología plantea un conjunto de prácticas que guían su aplicación [29, 30]:

- Programación en parejas
- Propiedad colectiva
- El juego de la planificación
- Entregas pequeñas
- Metáforas
- Diseño simple
- Pruebas
- Refactorización
- Integración continua
- Semana de 40 horas
- Cliente *in-site*
- Estándares de programación
- Espacio de trabajo abierto
- Reglas justas

[3]Se denomina así a una descripción de una funcionalidad deseada para un producto en desarrollo. Es aceptado en equipos ágiles, que representan un elemento del proyecto sobre el que se deberá trabajar en forma conjunta con el cliente para su definición y posterior desarrollo.

Los equipos que implementan XP aplican estas prácticas para el desarrollo del software. El producto es desarrollado siempre por dos personas en una misma estación de trabajo, esta colaboración se complementa con la posibilidad de que cualquier miembro del equipo pueda editar el código de la solución (propiedad colectiva). La interacción constante con el cliente se ve reflejada en las entregas y la planificación de las mismas, priorizando aquellas funcionalidades que aporten mayor valor para el negocio. La refactorización del código, la eliminación de fragmentos duplicados, la utilización constante de pruebas, el seguimiento de estándares de codificación y un diseño simple son la base de la flexibilidad del producto para adaptarse a cambios sin mayores problemas y que constituya una vía de comunicación para el equipo. Es a través de estos elementos que la integración frecuente del software es posible.

2.3.2.3 Lean Software Development

Basada en el sistema de producción automotriz de la empresa Toyota, esta metodología tiene por objetivo la construcción de los elementos del producto que aportan valor para el negocio y conforman los objetivos del proyecto, eliminando los "desperdicios" de este proceso [31, 32].

Se fundamenta en una serie de principios, prácticas y herramientas que facilitan su ejecución, entre los que se incluyen [31, 32]:

- Eliminar los desperdicios
- Amplificar el aprendizaje
- Decidir lo más tarde posible
- Entregar el producto lo más rápido posible
- Empoderar al equipo de trabajo
- Construir con integridad
- Ver el todo

A través de estos conceptos, se enfatiza la entrega de valor al cliente en términos de un producto que cumpla con sus requerimientos, sin la implementación de funcionalidades adicionales o no solicitadas. Ante cada presentación de avances, el *feedback* del cliente sirve como insumo para la mejora constante de los procesos de desarrollo y la identificación temprana de problemas, facilitando la aplicación de soluciones. De esta manera, se contribuye en la reducción del tiempo de entrega del producto, sin embargo, para lograr esto se requiere de un equipo con libertad suficiente para auto-organizarse y tomar las decisiones que considere oportunas para garantizar la flexibilidad en el desarrollo. Con respecto a la calidad del producto, la retroalimentación frecuente por parte del cliente y de las pruebas, complementada por un conjunto de métricas que permiten evaluar los avances son de gran utilidad para lograr una visión integral del mismo y su evolución a fin de identificar desvíos y aplicar medidas correctivas que beneficien a ambas partes.

2.3.2.4 Test Driven Development

Metodología iterativa e incremental que tiene como característica distintiva que una funcionalidad se desarrolla a partir de una especificación de pruebas que deberá superar para considerarse implementada correctamente [33]. El esquema de trabajo a seguir se puede describir como: especificación de los *tests* a través de código, implementación del código de la funcionalidad que supere tales pruebas, una instancia de refactorización para aquellos casos en los que se deben corregir errores y para reducir o eliminar la duplicidad de código [34].

A partir de un requisito del cliente se escribirán: una especificación, una serie de ejemplos para su comprensión que se denominan también casos de aceptación en los que los resultados serían correctos. Sobre estos elementos se especifican los *tests* para comenzar. Posteriormente, la funcionalidad se desarrolla conforme a lo especificado previamente, esperando que supere las pruebas especificadas. La funcionalidad se considera correcta, pero no en su versión final dado que se podrían agregar requerimientos en el futuro siguiendo estos mismos pasos. Estas actividades de refactorización no podrán acotarse únicamente al código de funcionalidades sino que también afectará a los conjuntos de pruebas para mantener la correctitud de lo implementado [33, 34].

Siguiendo estos pasos la metodología aporta beneficios tales como: la implementación se centra en las necesidades del cliente, no se desarrollan funcionalidades extra, se minimiza el número de errores que el producto puede presentar en una instancia de integración o producción y se obtiene un producto con mayor margen de reusabilidad y flexibilidad para la adopción de cambios [34].

2.3.2.5 Kanban

Es una técnica de gestión de proyectos, también relacionada con la empresa japonesa Toyota para la gestión del trabajo en una línea productiva [35]. Su nombre hace alusión a la visualización del avance de un proceso o proyecto a través de tarjetas que se movilizan sobre un tablero [36]. En particular, para desarrollo de software, suele ser utilizada en conjunto con Scrum, dando lugar a una metodología diferente que se denomina Scrumban [37].

En lo que respecta a Kanban, individualmente se basa en tres principios fundamentales [36]:

- **Visualizar el trabajo y las fases del ciclo de producción**: el seguimiento e identificación de la etapa en la que se encuentran las tareas se realiza a través de un tablero. El mismo es útil para determinar las actividades en ejecución en un punto determinado conjuntamente a la persona a quien se encuentra asignada y su prioridad.
- **Determinar un límite para el trabajo en curso en un momento dado**: el concepto de WIP[4] define para cada fase el número de tareas que pueden ejecutarse en forma simultánea. El objetivo de este elemento es que el foco del trabajo no sea iniciar las tareas sino terminarlas.
- **Medir el tiempo en finalizar una tarea**: la metodología utiliza dos métricas para medir la velocidad del desarrollo de una tarea en particular. Por un lado el *lead time* considera el tiempo desde que se registra un requerimiento o tarea por parte del cliente hasta que se entrega su desarrollo o resultado; por otro lado la métrica de *cycle time* indica el tiempo de ejecución en sí de una tarea, siendo más cercana a una medida de rendimiento.

[4] *Work in Progress* por su sigla en inglés.

2.3.2.6 ScrumBan = Scrum + Kanban

El caso de esta metodología es el de un enfoque híbrido en el que se toman las mejores prácticas de Scrum y Kanban para conformar un marco de trabajo en el que se optimizan procesos y actividades. La gestión de equipos, las iteraciones de corta duración, la estimación y priorización de las historias de usuario en los artefactos definidos (*backlogs*), las estrategias para revisión de los avances en el producto y la posibilidad de mejora continua son los elementos seleccionados desde la metodología Scrum. Mientras que en el caso de Kanban, se utilizan: la visualización y comunicación del flujo de trabajo, la organización del mismo, las limitaciones para las tareas en ejecución simultánea y las métricas para medir los avances en el proyecto [38, 39].

Entre las características principales de la metodología se pueden mencionar [38, 39]:

- No se definen sprints.
- Las reuniones de Scrum pueden usarse bajo demanda.
- El avance en el trabajo se observa desde el tablero y los gráficos de Scrum.
- Se mantiene la menor cantidad posible de historias de usuario en el tablero para evitar esfuerzos redundantes ante cambios en el contexto.
- Entre otros.

Estas características hacen que la metodología pueda mantener los objetivos de flexibilidad ante cambios, mejora continua de los procesos de trabajo, disminución de la carga de trabajo asociada a reuniones (ya que se realizan bajo demanda - necesidad) y el establecimiento de frecuencias específicas de revisión. ScrumBan se recomienda para proyectos que cumplen con alguna de las siguientes características: carga de trabajo bajo demanda, alta posibilidad de cambios en el contexto del negocio y/o necesidad de implementar flujos de trabajo con características especìficas [38].

2.3.2.7 Heart of Agile

Se trata de una metodología que plantea que la agilidad se ha complejizado desde la publicación del manifiesto ágil y propone retomar los principios básicos que la sustentan [40]. Esto lo plantea a partir de un conjunto de conceptos que se deben aplicar en la ejecución de un proyecto de desarrollo [40, 41]:

- **Colaboración**: el trabajo en equipo permite generar y desarrollar mejores ideas desde un inicio.
- **Entrega**: en un inicio del proyecto se presentan pequeños avances que permiten tener una mejor comprensión del dominio del producto. A medida que se avanza en el proyecto se podrá predecir la efectividad de las entregas en los resultados de la entidad.
- **Reflexión**: frecuentemente a lo largo de la ejecución del proyecto. Se debe aprender de la colaboración entre los miembros del equipo y a partir de las entregas que han sido evaluadas por el cliente. Este *feedback* soporta tanto al proceso como al producto.
- **Mejora**: relacionado con los puntos previos, son las oportunidades de optimizar cuestiones técnicas o de los procesos internos del equipo.

La metodología se presenta como una alternativa frente a otros esquemas de aplicación de técnicas ágiles en proyectos a gran escala dado a que se presenta, no desde la óptica de un conjunto de técnicas, sino a partir de cambios actitudinales y de comportamiento que permitirán obtener un mejor desempeño [40].

2.3.2.8 3x: Explore, Expand and Extract

Se trata de un modelo de trabajo para el desarrollo de productos, aplicable no solo al software, que se basa en tres fases claramente diferenciadas entre sí. Estas etapas son por las que pasa un producto en su desarrollo y explotación, y cada una tiene cuestiones particulares a considerar para el éxito de una iniciativa en desarrollo [42]. A continuación se describen las tres fases [43]:

- **Explore**: en la fase de descubrimiento de las funcionalidades del producto, el objetivo debe estar en la experimentación constante. Los desarrollos parciales se centran en la generación de soluciones al problema original y la evaluación del impacto que las mismas podrían tener tanto en la satisfacción del usuario como en el negocio. La calidad del producto no es un objetivo de alto valor en este punto.

- **Expand**: llega el punto en el que una o más de las soluciones planteadas en la fase anterior tiene éxito y comienza a ser utilizada de forma más "masiva". Entonces el objetivo del equipo debe estar en sostener ese crecimiento solucionando los problemas que puedan limitarlo. No se busca integrar nuevas funcionalidades sino que se busca brindar estabilidad a las ya disponibles a través de la resolución de problemas que pudieran detectarse.

- **Extract**: en esta fase se considera que se cuenta con un producto con relativo éxito y que empieza a presentar beneficios para el negocio. Es en este punto en el que se pueden incorporar más funcionalidades dado que la calidad del producto pasa a ser más importante, al mismo tiempo que se deben optimizar los procesos de trabajo sobre el software dado que su estabilidad debe ser garantizada.

En general, las tres fases no se solapan dadas sus características, pero sí podrían presentarse dentro de un mismo proyecto de desarrollo con diferentes funcionalidades del producto que van alcanzando niveles de utilización e impacto con mayor velocidad que otras. La fortaleza del modelo planteado se basa en la diferenciación de las tres etapas y su aplicación para el ciclo de vida de un producto, diferenciando prácticas y estilos de gestión para cada uno de los momentos descriptos.

2.3.2.9 Modern Agile

En esta metodología se plantea una modernización de las prácticas ágiles que han evolucionado desde su concepción y otras nuevas que han surgido en el último tiempo [44]. Toma como base fundamental cuatro principios que pueden ser implementados en organizaciones de diversos niveles para simplificar los procesos de desarrollo y la burocracia general de los mismos. Entendiendo que esta burocracia se presenta en forma

de un gran número de herramientas, *frameworks* pensados para ganar escalabilidad y certificaciones que no agilizan las prácticas en realidad [45]. La metodología no presenta roles, responsabilidades estrictas o prácticas específicamente recomendadas para un proyecto, su base consta de los cuatro principios que se describen a continuación [44, 45]:

- **Hacer que las personas sean geniales**: el foco del producto y su desarrollo debe estar en la entrega de beneficios al usuario para que el mismo alcance mejores resultados en sus tareas a través del uso del producto. La metodología no se limita solo a los usuarios sino a su ecosistema ya que el equipo involucrado en el proyecto de desarrollo debe tener todas las condiciones necesarias para poder ser "genial" en su trabajo.

- **Hacer de la seguridad un prerequisito**: en este caso se trata de un principio relacionado a la cultura organizacional en donde se deben aceptar los cambios. Cada miembro de un equipo puede realizar propuestas para incrementar la funcionalidad y éxito del producto en desarrollo. Ante los errores que puedan presentarse, la recomendación es aprender rápido de ellos para lograr optimizar los procesos de trabajo y mejorar la calidad del producto.

- **Experimentar y aprender rápido**: los principios anteriores se relacionan con poder experimentar con pequeñas propuestas de solución sin miedo a fallos, esta actividad y su frecuencia permiten aprender y mejorar los productos en desarrollo de manera más rápida. Los experimentos se vuelven parte de una estrategia de entrega de valor continua como marca el próximo principio.

- **Entregar valor continuamente**: un producto que no se encuentra a disposición del usuario no lo ayuda a cumplir con sus objetivos. Además, la entrega se vuelve más compleja conforme se agregan características, por lo que se propone dividir al mismo en partes que sumen valor progresivamente a medida que se encuentran disponibles. Este principio requiere de la aplicación de ciertas técnicas y herramientas que permitan automatizar el despliegue del software a fin de que los desarrolladores puedan realizar esta operación con frecuencia. Los productos a los que se hace referencia en este punto no necesariamente son funcionalidades nuevas sino que pueden incluirse mejoras leves o correcciones que se presentan a

un grupo reducido de usuarios para obtener *feedback* que permita la evolución del proyecto.

A modo de conclusión, la metodología plantea que la interacción con los usuarios es de gran utilidad para determinar si el esfuerzo que se está realizando en el desarrollo del producto es adecuado para solucionar los problemas objetivo del proyecto. A partir de esto, propone generar entregas parciales y frecuentes que sean evaluadas y sus aprendizajes incorporados en la gestión de la evolución del producto para lograr así un resultado de mayor calidad.

2.4 Métodos ágiles para proyectos de ciencia de datos

En los últimos años se han publicado diferentes metodologías o modelos de procesos que integran prácticas de metodologías ágiles para la gestión de proyectos de minería de datos / explotación de información / ciencia de datos. En las próximas secciones se reseñarán las características de algunos de estos enfoques y opiniones por parte de la industria al respecto de su utilización.

2.4.1 Prácticas ágiles en proyectos de ciencia de datos

La adaptación de prácticas ágiles a proyectos de ciencia de datos se ha dado por la trascendencia de las primeras y el impacto actual de los segundos. Inclusive se han generado iniciativas en las que se definieron una serie de principios aplicables que guardan algún tipo de equivalencia con el manifiesto ágil para el desarrollo de software [20].

En términos generales, los conceptos principales que se deben adaptar pueden resumirse de la siguiente manera [46]:

- **Entregables**: en estos casos en los que existe un componente de investigación, prueba y error, al definir cada iteración se deben mantener en términos alcanzables los productos a generar para no establecer una expectativa que no se pueda cumplir de cara al cliente.

- **Calidad de datos**: sin un equivalente directo en un proyecto de desarrollo de software, se trata de un aspecto a considerar ya que es esencial para la calidad de los resultados a obtener.
- **Comunicación**: los avances a presentar al finalizar cada sprint deben ser organizados en forma diferente dadas las características del producto a generar (conocimiento). El *feedback* del cliente a medida que se obtienen resultados intermedios debe permitir conocer su opinión para orientar las tareas de los siguientes sprints.

Con respecto a un equivalente al manifiesto ágil adaptado a proyectos de ciencia de datos se pueden encontrar tanto en la literatura académica como en publicaciones derivadas de la industria, dos propuestas que se reproducen a continuación:

- En el primer caso se trata de una serie de principios a tener en cuenta para ejecutar un proyecto de explotación de conocimiento aplicando conceptos ágiles [47, 48]:
  - Iteración constante y en todos los elementos del proyecto.
  - Cada iteración debe generar un entregable, aún cuando no esté terminado, ya que constituyen el medio para obtener retroalimentación por parte del cliente.
  - Dada la naturaleza de los problemas en este tipo de proyectos, la experimentación debe ser constante y junto al prototipado de soluciones tener más importancia que la culminación de tareas.
  - En las instancias de evaluación se integra lo que se denomina "la opinión de los datos" que se obtiene a partir del análisis de resultados parciales que pueden indicar la necesidad de cambios en futuras iteraciones.
  - Convertir a los datos crudos en instrumentos que aporten visualización y comprensión sobre los problemas en análisis. Y, a partir de ello, obtener predicciones que permitan brindar soporte a acciones a ejecutar.
  - Descubrir a través de las iteraciones y la experimentación aquellas partes del problema que puedan tener más relevancia y las soluciones que mejor se adapten a las mismas para alcanzar el éxito general del proyecto.
- En segunda instancia se puede encontrar una aproximación más similar al manifiesto ágil para desarrollo de software en lo que respecta a su redacción y principios asociados [49].

*"**Productos mínimamente viables** sobre prototipos,*

***APIs**[5] sobre bases de datos,*

***Un uso inteligente de los recursos computacionales** por sobre asunciones convenientes,*

***Tableros de control** sobre reportes,*

***Validación, seguridad y repetibilidad** sobre convenciones y costumbres."* [49].

De manera análoga al caso del manifiesto para el desarrollo de software, si bien se reconocen los elementos de la derecha, son más valorados los mencionados a la izquierda. Los principios que se recomienda aplicar se pueden resumir a los siguientes [49]:

- Automatizar lo más posible el procesamiento de datos.
- Concentrarse en la resolución de los problemas del proyecto, no en los métodos o algoritmos a emplear.
- Los resultados que se obtengan deberán ser evaluados, monitoreados y su generación automatizada.
- Se deben utilizar métricas que permitan evaluar la calidad del producto generado.
- Las tareas de recolección, integración y transformación de datos potencialmente requieren de una carga de trabajo relacionado con investigación de los enfoques de mayor utilidad. Estas acciones necesariamente pasarán a formar parte del trabajo de las iteraciones en donde se aborden tales fases del proceso.
- Los modelos intermedios que puedan obtenerse deberán ser evaluados para determinar su utilidad a pesar de no ser completos en vez de ser desechados a medida que se avanza en la generación de los modelos finales.

En general, en ambos casos se aplican visiones de agilidad que involucran conceptos para la gestión de los proyectos de ciencia de datos. A nivel general las iteraciones, la entrega constante de resultados y la evaluación de los mismos para determinar

[5] *Application Programming Interface* por su sigla en inglés. Se refiere a un conjunto de operaciones que un software disponibiliza para su uso por parte de otros productos.

su validez y calidad para la resolución de los problemas planteados son acciones que se repiten en los planteos. Por este motivo, se considera que los conceptos mencionados deberían considerarse más allá del nombre puntual que se otorgue al enfoque de aplicación de técnicas ágiles en un proyecto de este tipo.

2.4.2 ASUM-DM: Analytics Solutions Unified Method for Data Mining

Es una guía derivada de CRISP-DM que se orienta a implementar una solución de análisis de datos con principios ágiles. Su objetivo es acelerar la entrega de valor y minimizar riesgos en este tipo de iniciativas. Su composición es similar a CRISP-DM al estar organizada en: etapas, actividades de desarrollo, roles, responsabilidades, plantillas y guías de trabajo [50]. La extensión de la metodología original se fundamentó en la falta de detalle que presenta la original para la fase de implantación del proyecto.

Entre las características principales de ASUM-DM se destacan [50, 51]:

- Se encuentra ligada a las herramientas software de IBM Analytics, aunque se podría utilizar sin el soporte de tal plataforma.
- Se crea el rol de "equipo de gestión del proyecto" que asume la responsabilidad de supervisar la ejecución del mismo.
- Se utilizan diversos conceptos derivados de métodos ágiles como: iteraciones, participación del cliente en el equipo de desarrollo, prototipado de soluciones, instancias de mejora continua, instancias de pruebas sobre los resultados obtenidos, entre otros.

Por otra parte, las fases de la metodología se organizan de otra manera para reducir su cantidad de seis a cuatro más una [50]:

- **Análisis**: se considera equivalente a la fase de comprensión del negocio de CRISP-DM, tiene por objetivo determinar los requisitos y necesidades del cliente para obtener los objetivos del proyecto, estimando las primeras soluciones que se podría desarrollar.

- **Diseño**: en términos de la comparación con la metodología original abarca desde la comprensión de los datos hasta el modelado, por lo tanto se cubren los aspectos del desarrollo de la solución de minería de datos y análisis de datos.
- **Implantación**: en este caso es la fase en la que se ha puesto mayor atención (y presenta más cambios con CRISP-DM) para lograr que los resultados obtenidos con un proyecto de estas características no interfieran con la operatoria normal de la organización.
- **Operar y optimizar**: se trata de la fase en la que se controla la evolución del proyecto a lo largo de sus iteraciones para incorporar mejoras y/o ajustes que permitan cumplir con los objetivos definidos.
- **Gestión del proyecto**: se trata de una fase extra, paralela al resto, que agrupa las diferentes funciones que en un proyecto ágil basado en Scrum estarían a cargo de un Scrum Master. Estas actividades tienen por objetivo administrar y monitorear los procesos del proyecto.

2.4.3 TDSP: Team Data Science Process

Este modelo de procesos es planteado por la división de ciencia de datos del producto para la nube de la empresa Microsoft, Azure. Se presenta como un ciclo de vida para este tipo de proyectos en los que se tiene por objetivo la creación de modelos predictivos sobre algún tipo de escenario. Por lo mencionado, no es totalmente aplicable en proyectos destinados a otras operaciones dentro de lo que actualmente se considera "ciencia de datos" tales como: análisis exploratorio de datos, acciones vinculadas a la visualización de datos complejos y proyectos de análisis *ad-hoc* [52].

El modelo en general se compone por los siguientes elementos [52]:

- Un ciclo de vida para proyectos de ciencia de datos.
- Una estructura base para la definición y organización de los proyectos.
- Infraestructura y recursos para el desarrollo y despliegue del proyecto y sus resultados.

- Herramientas y utilidades de soporte para la ejecución del mismo.

El ciclo de vida planteado por la metodología guarda relación, a alto nivel, con lo definido en otros modelos como CRISP-DM o KDD, brindando un enfoque iterativo a través del cual se implementan modelos que pasarán a formar parte de aplicaciones de aprendizaje automático [53]. Se definen los objetivos, la forma de ejecutar la fase y los artefactos que se espera obtener al finalizarla. Tales definiciones para cada etapa del proceso favorecen la comunicación en el equipo de desarrollo y la identificación de las actividades a realizar. Las fases del ciclo de vida se listan a continuación [53]:

- Conocimiento del negocio.
- Adquisición y comprensión de los datos.
- Modelado.
- Implementación.
- Aceptación del cliente.

La característica extra que aporta la definición del ciclo de vida de TDSP radica en que plantea la integración de prácticas ágiles para la gestión del proyecto. Esto lo logra a través del uso de *sprints*, *backlogs* con historias de usuario, tareas e instancias de revisión de avances, entre otros elementos definidos en las herramientas de gestión de proyectos disponibles en Azure bajo la categoría de *Agile*.

El segundo de los elementos de la metodología es una estructura común para el proyecto, la misma tiene sentido al considerarse junto a plantillas para los documentos del proceso y la integración del trabajo de sus integrantes mediante algún sistema de control de versiones. La estandarización de la estructura del proyecto colabora con la lectura y ubicación de sus productos intermedios, teniendo directorios específicos para cada fase del proceso de desarrollo de la solución: documentación de los problemas a resolver, informes sobre las propiedades de los datos, fuentes correspondientes a la generación de modelos, documentos de evaluación de resultados, entre otros.

Con respecto a la infraestructura y recursos para la ejecución del proyecto, el modelo planteado recomienda ciertas prácticas relacionadas a la organización de la

disposición de los datos para el equipo y de las instancias de proceso. En general se hace referencia a sistemas de almacenamiento de datos, código, nodos de procesamiento de datos o servicios de ejecución de métodos de aprendizaje automático u otro tipo de técnicas aplicables, flujos de operaciones de integración y/o despliegue continuo utilizando contenedores, entre otros. Para ello, establece algunas recomendaciones basadas en los productos disponibles en la nube de Microsoft, Azure, en su división para *Machine Learning*.

Finalmente, en el apartado de herramientas y utilidades se proveen recursos destinados a la aceleración de los procesos dentro del proyecto mediante la provisión de código para la automatización de ciertas tareas como: exploración de datos, visualización y modelado básico. Estos recursos podrán ser adaptados en función de los requisitos del proyecto en cuestión, simplificando ciertas tareas para un progreso más rápido. Sin embargo, algunas de las herramientas están implementadas sobre la nube de Microsoft por lo que es un punto a analizar por los equipos que consideren utilizar la metodología.

2.4.4 Scrum y ciencia de datos

Si se toman las dos metodologías o marcos de trabajo más difundidos tanto de desarrollo de proyectos de minería de datos, CRISP-DM, y de desarrollo ágil de software, Scrum, se puede establecer un enfoque híbrido que las unifique para un proyecto de ciencia de datos. En primer lugar, el esquema de trabajo implementado en Scrum se compone de ciclos de desarrollo incremental en los que se observa una estructura que puede definirse como PER3[6] [54]. Los pasos de esta estructura se describen a continuación:

- **Planificar**: se definen los items de trabajo listos para ser incorporados en una iteración, en la práctica son aquellos que cuentan con la DoR[7]. Usualmente las acciones de esta etapa se realizan en la reunión de planificación del sprint.

[6] *Plan, Execute, Review, Retrospective and Refinament* por su sigla en inglés.
[7] Definition of Ready por su sigla en inglés. Son los items de trabajo que han sido revisados por el cliente y los miembros del equipo de trabajo y pueden ser desarrollados en un *sprint*.

- **Ejecutar**: a lo largo de una iteración se desarrollan las acciones necesarias para que cada item de trabajo alcance la DoD[8]. Además se incluye el seguimiento en las reuniones diarias del equipo de desarrollo.
- **Revisar**: a medida que las iteraciones generan evoluciones de uno o más productos del proyecto, las mismas son evaluadas desde la óptica de todos los involucrados en la reunión de revisión del *sprint*.
- **Retrospectiva**: de igual manera, el trabajo realizado por el equipo en cada iteración es analizado a nivel de la gestión para optimizar procesos y detectar cuestiones a corregir en próximas iteraciones. Tales elementos pasan a conformar lo que se suele denominar deuda técnica.
- **Refinamiento**: los items de trabajo que se encuentran en el *product backlog* a medida que pasan los *sprints* podrán ser refinados en términos de especificar más detalles en su definición o integrar aclaraciones que simplifiquen su comprensión al momento de ser desarrollados.

Estas características de Scrum, al adaptarlas a las fases de CRISP-DM resultan en la siguiente distribución [55, 56]:

- Las fases de Comprensión del negocio y Comprensión de los datos se ejecutan a través de la confección del *product backlog* y el *sprint backlog*.
- Las fases de Preparación de los datos, Modelado y Despliegue son ejecutadas en cada sprint del proyecto. De esta manera, cada item de trabajo que se determina abordar en una iteración pasará por las tres etapas mencionadas.
- La fase de Evaluación se relaciona directamente con la reunión de revisión de los resultados de un sprint en donde se evalúan los incrementos desarrollados.

A través de lo mencionado previamente, en sucesivas iteraciones, diferentes items de trabajo del proyecto serán desarrollados y generarán avances con respecto a brindar una solución a los objetivos del mismo. La gestión de las tareas de CRISP-DM a través de los eventos y artefactos de Scrum permiten la aplicación de un enfoque

[8] *Definition of Done* por su sigla en inglés. El estado o conjunto de requisitos que debe cumplir un elemento que se esté desarrollando para determinar que ha sido finalizado para el equipo de trabajo.

iterativo e incremental para un proyecto de ciencia de datos en lugar de uno tradicionalmente descripto como de tipo cascada. Sin embargo, este planteo no se encuentra exento de cuestiones a resolver a fin de lograr una implementación exitosa. Algunas se detallan a continuación [55, 57, 58, 59]:

- **Determinar los incrementos a generar en cada sprint**: no necesariamente serán modelos completos para el problema en cuestión sino que podrían ser productos intermedios como un *dataset* sobre el que se hayan aplicado técnicas de preproceso, reportes de calidad de datos, reportes de metadatos, entre otros. Independientemente del elemento en sí, la posibilidad de contar con una visión del avance del proyecto y la retroalimentación por parte del cliente es fundamental para la gestión del proyecto.
- **Determinar cómo se va a gestionar la integración**: abarcando las fuentes de datos involucradas, su integración y la aplicación de transformaciones para lograr homogeneidad en el formato de los datos para su posterior procesamiento. Estas actividades podrían ser distribuidas en más de una iteración a fin de analizar sus resultados parciales y buscar un *dataset* de la mayor calidad posible.
- **Determinar la conformación del equipo de trabajo**: el carácter multifuncional y auto-organizado de los equipos de Scrum debe mantenerse en este enfoque. Si bien puede haber perfiles con un mayor grado de especialidad en algunas tareas, el objetivo en estos casos será minimizar los tiempos de espera entre tareas que sean dependientes entre sí, además de optimizar la comunicación para que el producto responda a los requerimientos planteados.

A modo de resumen, Scrum provee un enfoque para proyectos de ciencia de datos que podría permitir la adaptación a entornos cambiantes tales como los que pueden caracterizar a este tipo de iniciativas. Por otra parte, la inclusión del cliente en el equipo de trabajo y la utilización de iteraciones de duración fija permitiría obtener mejores resultados y en plazos reducidos que favorezcan la evaluación del producto generado (en forma parcial o completa) y contribuyan a la calidad final del desarrollo [55, 56, 57].

Capítulo 3

Descripción del problema

En este capítulo se reseña el problema a resolver en este trabajo final de maestría. En primera instancia se describen las generalidades del caso para posteriormente abordar detalles sobre la investigación realizada junto a sus alcances y limitaciones.

3.1 Generalidades

En general, las metodologías o modelos de gestión de proyectos de ciencia de datos mencionados en el capítulo anterior se enfocan en la aplicación de las técnicas y/o algoritmos sobre los datos disponibles para extraer conocimiento. De esta manera, no se realiza un análisis en profundidad de los problemas a resolver desde una perspectiva de negocios ni se abordan, generalmente, cuestiones relacionadas a la gestión de tales iniciativas. Mientras que en el ámbito del desarrollo de software se han generado e implementado con éxito diversas metodologías, técnicas y procesos considerados ágiles en torno a la gestión. La adaptación al cambio, la entrega constante de valor, la colaboración con el cliente y la organización del trabajo en iteraciones son parte de los pilares de este enfoque de trabajo.

El acercamiento de las técnicas de gestión ágil a los proyectos de ciencia de datos no es nuevo, existiendo desarrollos como una propuesta de manifiesto y marcos de trabajo adaptados como los presentados previamente. Sin embargo, las alternativas disponibles presentan algunos inconvenientes que hacen necesario plantear una solución

que contemple, entre otras cosas, las herramientas necesarias para brindar soporte a las prácticas seleccionadas y los elementos de adaptación para organizaciones o problemas de un menor tamaño que cuentan con una realidad diferente a las grandes empresas. Además de solventar algunos problemas latentes que pueden afectar el desarrollo de las iniciativas y generar resultados no esperados.

En este contexto, el objetivo del presente trabajo consiste en seleccionar técnicas aplicables a la gestión de proyectos de ciencia de datos que permitan obtener beneficios tales como los presentados para la ingeniería de software. La propuesta de técnicas y su marco de utilización se complementa con una selección de herramientas a emplear como soporte y un conjunto de aspectos a adaptar para su aplicación en entornos organizacionales de tamaño reducido. En este sentido, se busca poner el foco en la visibilidad y trazabilidad de las acciones a ejecutar, junto a la generación de productos de datos que en las sucesivas iteraciones aporten valor al contexto de negocios del problema.

3.2 Objetivos

El objetivo general del presente trabajo es:

- Relevar y seleccionar un conjunto de técnicas derivadas de métodos ágiles de gestión de proyectos para su aplicación en un entorno de ciencia de datos en pequeñas y medianas organizaciones.

Entre los objetivos específicos planteados se encuentran:

- Relevar, analizar y seleccionar los procesos, etapas y procedimientos que involucra la ejecución de un proyecto de ciencia de datos.
- Relevar y analizar métodos y técnicas derivados de metodologías ágiles, principalmente provenientes del ámbito de la ingeniería de software, para la gestión de proyectos.
- Analizar las actividades de gestión que involucran los diferentes procesos o etapas de un proyecto de ciencia de datos y la aplicabilidad de las técnicas resultantes del relevamiento previo.

- Seleccionar un conjunto de técnicas ágiles para su aplicación en las diferentes actividades de gestión de un proyecto de ciencia de datos.
- Seleccionar un conjunto de herramientas software de acceso abierto que brinden soporte a las técnicas seleccionadas.
- Ejecutar una instancia de validación de la selección propuesta sobre un caso de estudio a fin de validar su aplicabilidad.
- Publicar los resultados de la investigación conducida.

3.3 Alcance y limitaciones

El trabajo final de maestría cubre la selección de una serie de técnicas de gestión derivadas de modelos o metodologías ágiles para su implementación en proyectos de ciencia de datos. Se adjunta una propuesta de esquema de trabajo o plantilla para el desarrollo de algunas actividades que requieren una adaptación debido a las particularidades del contexto en cuestión. Por otra parte, se establecen criterios para la selección de herramientas de soporte para la ejecución de tales acciones siguiendo, en este aspecto, las recomendaciones de la industria. En todo momento, se tiene en cuenta una serie de componentes adaptables para que la propuesta pueda ser utilizada en el tipo de organizaciones previsto.

Entre las limitaciones del trabajo se pueden mencionar:

- No se trata de una metodología en sí misma, sino que se ha generado un marco de trabajo consistente en técnicas de gestión aplicables en diferentes instancias del desarrollo de un proyecto de ciencia de datos.
- No se recomiendan herramientas software específicas, brindando espacio para la adaptación al contexto de cada organización.
- No se presentan estrategias para escalar la propuesta a grandes organizaciones sino que se plantean como una línea de trabajo a desarrollar.

- No se detallan ciertas prácticas metodológicas tanto de gestión como de índole técnico para ciencia de datos dado que cada organización podría adaptar las disponibles en el mercado o las que utilice previamente.

Capítulo 4

Solución propuesta

En este capítulo se presentan los aspectos relativos al desarrollo de la solución definida para el problema descripto en el capítulo anterior. Se inicia por una introducción general del contexto en el que se plantea la misma, las diferentes técnicas de gestión aplicables, la solución en sí y, como parte de la misma, la selección de herramientas software recomendadas para brindar soporte a la implementación del proceso en un proyecto de ciencia de datos.

4.1 Introducción

La gestión ágil de proyectos cuenta con modelos de procesos y metodologías que emplean diferentes técnicas para manejar cada etapa de un proyecto. Sin embargo, el uso que se realiza de estas técnicas no es necesariamente equitativo en todos los ámbitos y equipos, sino que la idea de adaptación para cubrir las necesidades particulares de cada contexto genera que algunas sean más utilizadas que otras [60]. El mayor grado de difusión y adopción de los métodos ágiles se ha dado en el mundo del desarrollo de software. En los proyectos de ciencia de datos, donde conviven la incertidumbre en torno al entendimiento del problema y sus posibles cambios, las diferentes alternativas de solución y el caracter iterativo de su desarrollo, la aplicación de un esquema de gestión ágil se presenta como una opción válida tanto en la literatura del área como en reportes de experiencias en la industria [61, 62, 63].

Los conceptos a aplicar en la gestión ágil de un proyecto de ciencia de datos abarcan básicamente a los siguientes [61]:

- Usar iteraciones cortas
- Adaptarse a los cambios
- Obtener retroalimentación en cada iteración
- Entrega constante de valor

La organización del trabajo en iteraciones cortas permite una mayor capacidad de adaptación al cambio y la generación de resultados en forma constante. Aún si estos resultados no fueran del todo "positivos", conformarán parte del conocimiento del dominio del problema que permitirá priorizar y estimar con menor incertidumbre futuras iteraciones. La retroalimentación del usuario / cliente sobre el trabajo realizado por el equipo aporta beneficios en este sentido para orientar los esfuerzos a realizar a futuro en aquellos elementos que agreguen valor al negocio. Estos aspectos se adaptan a la naturaleza de los proyectos de ciencia de datos en los que se cuenta con una gran incertidumbre, presentan una progresión no linear y suelen ajustar sus objetivos en función de una oportunidad o necesidad del negocio, por lo que este tipo de organización del trabajo se considera un enfoque más que válido [61, 62, 64].

Según el tipo de proyecto del que se trate y los objetivos que se hayan planteado sobre el mismo, los beneficios de integrar un enfoque ágil en su gestión podrían relacionarse con la puesta en marcha de un proceso de entrega continua de resultados [64]. De esta manera, el desarrollo de un prototipo de solución que determine la viabilidad de la iniciativa junto a la implementación de una versión inicial de la solución (un mínimo producto viable, MVP según su sigla en inglés) obedece a este requerimiento de adaptación y entrega de valor constantes. Esto simplifica que, en caso de que el proyecto requiera de un mayor margen de trabajo orientado a la investigación de algún aspecto del problema, el cliente / usuario previamente podría contar con una solución que, aunque parcial, sea de utilidad para sus objetivos [65, 66]. Además, en este planteo la retroalimentación de los usuarios en pos de la mejora constante de la calidad del producto en desarrollo conforma otro aspecto relevante para optar por un enfoque de gestión ágil para este tipo de proyectos [52, 62, 65, 66]. Finalmente, diferentes iniciativas

podrían utilizar enfoques de administración basados en la selección y adaptación de aquellas técnicas que se consideren adecuadas tanto para las características del problema como al equipo de trabajo que lo esté abordando [66].

En este contexto, el desarrollo de la solución propuesta para el presente trabajo implicó los siguientes pasos:

1. Relevamiento de técnicas o modelos de gestión de proyectos ágiles aplicables.
2. Relevamiento de enfoques de gestión de proyectos de ciencia de datos que aplicaran agilidad.
3. Elaboración de la propuesta de marco de trabajo que constituye el objetivo principal del trabajo.
4. Selección y recomendación de herramientas software para dar soporte al enfoque metodológico propuesto.

En las siguientes secciones del presente capítulo se abordan estos pasos.

4.2 Relevamiento de técnicas ágiles aplicables a proyectos de ciencia de datos

Para comenzar con este relevamiento se determinó recurrir directamente a quienes utilizan en su día a día técnicas y/o metodologías ágiles al menos desde una perspectiva ligada al desarrollo de software. Esto se ha obtenido principalmente de encuestas que son realizadas año a año en forma internacional y abarcan a un gran número de organizaciones de diferentes tamaños y objetivos, con lo que se considera una muestra adecuada para los fines del presente trabajo [60]. En la continuidad de la presente sección se mencionarán: los principales objetivos y beneficios que se buscan a partir del uso de un enfoque ágil, las metodologías y prácticas tanto de gestión como índole técnico que son más utilizadas y las herramientas de soporte que se relacionan adecuadamente a los items precedentes.

En primer lugar, entre los motivos y beneficios de utilizar un enfoque ágil se destacan [60]:

- Acelerar la entrega y disponibilidad de resultados.
- Habilidad para adaptarse a los cambios.
- Mejorar la productividad del equipo de trabajo.
- Reducir los riesgos del proyecto.
- Alinear los objetivos del negocio y los de tecnologías de la información.

En general se puede observar una tendencia a entender que el beneficio de la agilidad en la gestión reside en la adaptación a los cambios y que esto impacte de forma positiva en la entrega de un producto que aporte valor al negocio que desarrolle la organización. En función del despliegue de las soluciones relacionadas al proyecto en cuestión la productividad de un equipo será potencialmente mayor y todo el *combo* de este enfoque colabora para que los riesgos que se hayan detectado para el proyecto en cuestión puedan ser gestionados.

En cuanto a las metodologías o marcos de trabajo más utilizados se pueden mencionar [60]:

- Scrum con cerca de más de la mitad de la participación global.
- Scrum + XP o ScrumBan (Scrum + Kanban) con cerca de un 10% cada una.
- Métodos híbridos en donde se utilizan prácticas asociadas a más de una metodología o modelo también con cerca del 10%.

Este dato permite entender que Scrum es el marco de trabajo más empleado, tanto en un formato "puro" como en un enfoque híbrido en donde se asocia con diferentes metodologías como XP o Kanban. La relevancia de este tipo de casos en donde se adaptan las prácticas a utilizar no es menor. Además se debe marcar que el alto grado de uso de Scrum no diferencia entre los casos o proyectos en los que se usa todo el conjunto de reuniones, artefactos y prácticas propuestas con aquellos en los que se usa solo una parte de estos.

Con respecto a las prácticas de gestión, aquellas que han obtenido un grado de utilización mayor al 50% en los resultados de la encuesta han sido [60]:

- Reunión diaria del equipo de trabajo.
- Reunión de planificación de la iteración.
- Reunión de retrospectiva.
- Reunión de presentación de los resultados de la iteración.
- Iteraciones de corta duración.

Iteraciones cortas para tener una ventana de oportunidad de adaptarse a cambios en el contexto, eventos que aportan al trabajo en equipo y su mejora progresiva, como son la reunión diaria y la de retrospectiva. Además de las reuniones que cuentan con la participación del cliente / usuario del producto en desarrollo para lograr que los objetivos siempre sean claros y la prioridad la marque el negocio, haciendo que el proyecto sea capaz de aportar valor al mismo.

Mientras que en el apartado de las prácticas técnicas se destacan en una proporción similar [60]:

- Pruebas unitarias.
- Estándares de codificación.
- Integración continua.
- Refactorización.
- Entrega continua.

En este caso, se trata por un lado de cuestiones de base para la calidad del producto, pruebas y un acuerdo común de codificación que no abarca solo al código sino también a diversos aspectos de arquitectura y organización de la solución. La refactorización como un camino a seguir para los casos en los que se realiza una primera versión del producto o alguna parte del mismo y posteriormente se mejora. Y finalmente, prácticas que son de índole técnico pero que también reflejan la forma en la que se gestiona el proyecto, entrega continua para que el producto tenga un menor *time-to-market* (tiempo que tarda desde que inicia su desarrollo hasta que es utilizado por los

usuarios) e integración continua como base para lograr lo anterior y agilizar los procesos del pase de desarrollo a producción.

Por último, a nivel de herramientas genéricas que se utilizan a la ejecución de un proyecto ágil se pueden mencionar [60]:

- Tablero kanban.
- Pizarra de tareas (*taskboard*).
- Software para el seguimiento de incidencias.
- Herramientas relacionadas a la integración continua, pruebas unitarias y automatización de procesos de despliegue.
- Hojas de cálculo.

En todos los casos se trata de herramientas que servirán como soporte no solo para el aspecto de gestión del proyecto sino que podrían tener incidencia en las cuestiones más ligadas al producto y a su construcción mediante las iteraciones que se van ejecutando progresivamente. En este punto, es importante destacar que el listado anterior no hace referencia a una herramienta software en particular. Sin embargo, en la misma encuesta se relevan estos datos mostrando como favorita a la herramienta Jira de Atlassian [67], aunque también se presentan diversas alternativas de funcionalidades similares provenientes de otros vendedores como Azure Repos [68] / DevOps [69] (Microsoft). Tomando como referencia a estas dos herramientas software se puede intuir que el objetivo de su uso es el de tener en un mismo espacio a los requerimientos, su distribución en las diferentes iteraciones del proyecto y su descomposición en tareas técnicas. Así como también la vinculación al código del producto en desarrollo y una herramienta que permita el control y monitoreo de las acciones de integración y despliegue continuo de las diferentes versiones del mismo en caso de que tales acciones sean parte del proyecto.

Finalmente, a modo de cierre se consideraron las prácticas o técnicas listadas en la encuesta State of Agile [60] y estableció una relación sobre los conceptos presentes con otros materiales de referencia en la materia como son el Map of Agile [70] y el Agile Glossary [71] que genera la Agile Alliance. La relación de correspondencia entre

las técnicas y el modelo de procesos o metodología de origen de las mismas se puede observar en las siguientes tablas: la 4.1 muestra las prácticas relacionadas con la gestión del proyecto mientras que la 4.2 muestra las referidas a cuestiones técnicas.

A partir del contenido de ambas tablas se puede observar cuáles son las prácticas o técnicas tanto de gestión como de índole técnico aplicables en un proyecto ágil. Además se ha establecido una correspondencia con conceptos o prácticas con un mayor nivel de granularidad que permitirá realizar una selección más adecuada para las fases posteriores del plateo de la solución. Esto se puede observar con un simple ejemplo: en la realización de una reunión de planificación de una iteración intervienen diferentes elementos a considerar, entre ellos: la definición de listo, la vista del listado de historias de usuario pendientes del producto, la posibilidad de refinar algunas de ellas antes de una próxima iteración, también la posibilidad de dividirlas en nuevas historias de usuario de menor tamaño y la realización de una estimación de las mismas con respecto a su implementación.

4.3 Relevamiento de enfoques de gestión de proyectos de ciencia de datos que aplican agilidad

En el capítulo dos del presente trabajo se han abordado las principales metodologías o modelos de procesos para proyectos de ciencia de datos que han sido empleados desde el inicio del siglo XXI. En esa enumeración, la figura de CRISP-DM [15] se ha remarcado como el estándar *de facto* para este tipo de proyectos, condición que se mantiene en general hasta la actualidad aunque con algunas variantes. También se mencionó la propuesta inicial que diera origen al vocablo KDD [6] junto a SEMMA [14, 16] y Catalyst [18] como metodologías alternativas.

Establecida la base metodológica tradicional para este tipo de proyectos, se describieron iniciativas que buscaban integrar prácticas de gestión de tipo ágil en su desarrollo a modo de alternativa. En tal apartado se mencionó a ASUM-DM [50, 51] y TDSP [52] como dos planteos formalizados y potencialmente completos, tanto por la especificidad para este tipo de proyectos como por el nivel de detalle en la definición de sus actividades y fases. Contando, además, con la ventaja de que algunas de sus prácticas se encuentran integradas a herramientas software que simplifican su aplicación en el

State of Agile	**Map of Agile**	
Técnica / Práctica	**Concepto / Técnica / Práctica**	**Origen / Relación**
Reunión diaria	3 preguntas	Scrum
	Tablero de tareas	Scrum
	Timebox	Scrum
Sprint planning	Definición de Listo	Scrum
	Iteraciones	Scrum / XP
	Backlog	Scrum
	Historias de Usuario	XP / *Product Management*
	Refinamiento del *backlog* (*grooming*)	Scrum / *Product Management*
	Story Splitting	*Product Management*
	Story Points / Points estimates	Scrum
	Mapa de historias de usuario	*Product Management*
Retrospectivas	Retrospectiva	Trabajo en equipo
	Facilitadores	Trabajo en equipo
Sprint *review*	Definición de Hecho	Scrum
	Pruebas de aceptación	*Testing*
Iteraciones cortas	*Timebox*	Scrum
	Entrega frecuente	XP
	Iteraciones	Scrum / XP
Planning poker	*Planning poker*	Scrum
Kanban	Tablero de tareas	Scrum
	Tablero Kanban	Lean
Release planning	Mapa de historias de usuario	Product Management
	Entrega frecuente	XP
Cliente dedicado	Equipo completo	Trabajo en equipo / Fundamentos
Equipo único	Equipo completo	Trabajo en equipo / Fundamentos

Tabla 4.1: Prácticas de gestión y su relación con métodos ágiles.
Fuente: elaboración propia con datos de [60, 70].

State of Agile	**Map of Agile**	
Técnica / Práctica	**Concepto / Técnica / Práctica**	**Origen / Relación**
Test unitarios	Tests unitarios	*Testing*
	Mocks	*Testing*
Estándares de código	Reglas de simplicidad	*Design*
	Propiedad colectiva de código	XP
Integración continua	Integración continua	DevOps / XP
	Versionado	DevOps / Fundamentos
	Construcción automatizada	DevOps
Refactorización	Refactorización	XP / *Design*
	Versionado	DevOps / Fundamentos
	Tests unitarios	*Testing*
	Construcción automatizada	DevOps
Entrega continua	Entregas frecuentes	XP
	Entrega Continua (*Delivery*)	DevOps
Despliegue continuo	Despliegue continuo (*Deploy*)	DevOps
	Entregas frecuentes	XP
Programación de pares	Programación de pares	XP
	Test Driven Development	XP / *Testing*
TDD	*Test Driven Development*	XP / *Testing*
	Refactorización	XP / *Design*
	Reglas de simplicidad	*Design*
	Tests unitarios	*Testing*
Tests de aceptación	Tests de aceptación	*Testing*
	Acceptance Test Driven Development	*Testing*
Propiedad colectiva del código	Propiedad colectiva de código	XP

Tabla 4.2: Prácticas técnicas y su relación con métodos ágiles.
Fuente: elaboración propia con datos de [60, 70].

caso del modelo presentado por Microsoft a través de su servicio en la nube, Azure. Por otra parte, se mencionaron estrategias en donde se incorporan a la gestión del proyecto prácticas o técnicas específicas que buscan agilizar su gestión sin que esto diera lugar a una metodología o marco de trabajo en particular [48]. En estos casos, la utilización de Scrum se observó como un punto de partida común que se ha utilizado junto a diversas fases o actividades obtenidas de CRISP-DM para conformar una alternativa de gestión [54, 55, 56, 61].

Las diferentes estrategias relevadas de este último grupo mencionado, si bien mantuvieron en general la organización en fases de CRISP-DM, han modificado las actividades incluidas en cada una de ellas a fin de que la adaptación a una gestión de tipo ágil sea consistente. Es así como la fase de comprensión del negocio tiene una marcada importancia en la definición de los límites del proyecto y la priorización de actividades; las diferentes iteraciones incorporan elementos o actividades propias de las fases de comprensión y preparación de los datos junto a modelado. Mientras que las actividades de las fases de evaluación y despliegue se alternan según la propuesta analizada entre parte de la iteración o en acciones a ejecutar en forma posterior a la misma al estilo de una revisión de los resultados de la misma o en una instancia similar a una reunión de retrospectiva más propia de Scrum.

Existen otras propuestas denominadas "híbridas" que se detallan brevemente a continuación:

- **Scrum + CRISP-DM + PMBoK** [64]: en este caso se trata de un enfoque en donde, utilizando como base el ciclo de trabajo de Scrum, se alternan actividades de gestión originadas en el contenido del PMBoK [72] y actividades técnicas obtenidas desde CRISP-DM [15]. Consta de tres fases en donde se realizan el pre-proyecto, la definición de la visión y alcance del mismo; la preparación de los datos, el desarrollo de los modelos o productos de datos a generar a través de iteraciones; y finalmente se cierra el proyecto considerando diferentes acciones relacionadas a la aceptación de los resultados obtenidos por parte del cliente, además de delinear acciones de mantenimiento.
- **Scrum + CRISP-DM** [63]: tal como se describiera en la sección 2.4.4 en este caso se toma el enfoque de desarrollo iterativo de Scrum y en cada iteración se desarrollan actividades que se seleccionan desde el modelo planteado en CRISP-DM. Las

fases de comprensión del negocio y de los datos se equiparan a las actividades de generación de las pilas del producto y de cada iteración. En cada *sprint* se realizan actividades de preparación de datos, modelado y, parcialmente, de evaluación. Finalmente, en forma posterior al *sprint* se realiza la presentación del incremento del producto generado y una reunión de retrospectiva. Se incluye una fase o etapa extra para la realización del despliegue de la solución generada a una instancia de producción.

- **Agile KDD** [73]: esta propuesta nace de la unificación de las metodologías *Open UP* [74] de desarrollo de software y KDD [6] para ciencia de datos. La primera de ellas funciona como una estructura en la que se integran componentes de la segunda. Es así como se obtienen cuatro fases en las que se pasa de entender el dominio del problema y seleccionar el *dataset* objetivo; a diseñar la arquitectura de la solución y realizar la limpieza de los datos. Posteriormente se construyen los productos o modelos necesarios para la resolución del problema en cuestión junto a una evaluación de los mismos. Y finalmente se pasa a realizar el despliegue de la misma para que pueda ser utilizada por el cliente.

- **ASD-DM** [75]: sigla en inglés para *Adaptive Software Development on Data Mining*. Este marco de trabajo se presenta como una adaptación de la metodología de desarrollo de software adaptativa (ASD) para la generación de soluciones a problemas de ciencia de datos. En detalle, es similar a la mencionada en el punto anterior en el sentido que ASD funciona como estructura para las actividades estrictamente relacionadas al desarrollo del modelo o producto de datos en cuestión. Se trata de tres fases en las que se inicia por entender el negocio, a los datos y prepararlos; luego se busca obtener el mejor modelo (ya que el planteo es sobre minería de datos de tipo predictiva). Finalmente, se realiza el aprendizaje del proceso conducido, con la evaluación del modelo y su despliegue.

Como se puede observar en la enumeración previa, existen alternativas metodológicas que integran agilidad en la ejecución de proyectos de ciencia de datos. Cada una de ellas aborda el problema de la gestión de estos proyectos y sus problemas a resolver de diferente manera. Sin embargo, se pueden observar algunos puntos en común que también se encuentran en las experiencias relatadas por parte de diferentes actores de la industria [61, 65, 66]. Es así como, a partir del análisis de estas alternativas, se

decidió construir un listado de actividades de las metodologías de ciencia de datos que se pueden considerar equivalentes a la selección presentada en la sección anterior en el caso de prácticas ágiles.

4.3.1 Puntos en común entre las diferentes estrategias de integración de agilidad en ciencia de datos

En este apartado, se refleja el análisis realizado de las propuestas relevadas para la gestión de proyectos de ciencia de datos mediante un enfoque ágil. Se detallan las premisas o condiciones que, tanto en un ámbito académico como profesional o industrial, se tienen en cuenta para la ejecución y organización de las actividades de un proyecto. Tal como se mencionó previamente, el punto de partida relativamente común para todas las propuestas es el conjunto de fases de CRISP-DM, aunque con ajustes en torno a qué actividades se incluyen o no en cada fase. En el apartado de gestión, se puede observar alguna diferencia más marcada con iniciativas que utilizan o no ciertas técnicas o prácticas. El listado será dividido, entonces, en dos partes: las prácticas de tipo técnicas y las de gestión.

En relación a las prácticas técnicas abordadas en las propuestas se destacan [50, 52, 63, 64, 66, 73, 75]:

- Comprensión del negocio.
- Comprensión de los datos.
- Preparación de los datos.
- Modelado (o desarrollo de la solución).
- Evaluación.
- Despliegue.

Como se puede observar, guarda correspondencia con lo planteado por CRISP-DM. Sin embargo, existen algunos detalles a considerar:

- La comprensión del negocio y de los datos se realiza de tal manera que permita establecer un conjunto de métricas y umbrales objetivo para las mismas a fin de considerar la viabilidad del proyecto [52, 66, 76]. Estos indicadores serán evaluados una vez que se cuente con una versión funcional del producto de datos[1] a desarrollar que conforme una prueba de concepto (PoC por su sigla en inglés) [66, 76, 77, 78]. De esta manera ante un desarrollo cuyo *gap*[2] sea considerado insalvable por el equipo de trabajo se podrá decidir no proceder con el proyecto hasta tanto las causas de tal situación sean subsanadas.

- La fase de modelado o desarrollo es, naturalmente, común a todas las iniciativas ya que es en la que se realiza el trabajo necesario para la generación del modelo, la solución o cualquiera fuera el producto de datos objetivo del proyecto. Las variaciones se encuentran ligadas a la forma en la que tal actividad es ejecutada según el enfoque planteado, por lo que será detallada en el próximo conjunto de prácticas relacionadas a la gestión.

- La preparación de los datos, dadas las tendencias vigentes al momento de la ejecución del proyecto en cuestión, suele abarcar también una descripción o establecimiento inicial de la arquitectura de la solución a implementar. Entre los factores a considerar se podrían mencionar: el volúmen de datos a procesar, el tipo de información a generar, su esquema de visualización, la integración con otras soluciones dentro de la organización, entre otros [50, 52, 73].

- Con respecto al despliegue del producto, su puesta a disposición para los usuarios finales, se ha encontrado una tendencia a marcar como una necesidad el pase a producción de resultados en etapas tempranas del proyecto en vez de hacerlo cuando los valores de los indicadores de calidad de la solución sean óptimos [65, 66, 76, 77, 78]. Esto se relaciona con una tendencia en los proyectos de ciencia de datos a realizar sucesivas tareas de investigación para mejorar tales indicadores. Y, en consecuencia, esperar para pasar de una instancia de laboratorio o experi-

[1] Se utilizará esta denominación para englobar a los productos que, haciendo uso de datos, podrían resolver uno o más de los problemas de un proyecto. Se incluye en la definición a herramientas de visualización, modelos de predicción, métodos de recomendación, sistemas de soporte a la toma de decisiones, entre otros.

[2] Se denomina así a la diferencia entre dos valores, en este contexto es la diferencia entre el valor obtenido mediante la PoC en los indicadores definidos y los valores esperados para los mismos que se establecieron al inicio del proyecto.

mentación a una de producción a la solución, limitando la entrega de valor por parte del proyecto a la organización.

En cuanto a las técnicas de gestión, se pueden mencionar a las siguientes como las más recurrentes:

- Planificación de iteraciones y versiones [63, 64, 65, 66, 73].
- Reunión de retrospectiva [62, 64, 65, 66, 77].
- Reunión de demostración de avances de la iteración (demo / *sprint review*) [62, 65, 66, 76, 77].
- Definición de los entregables por iteración / item de trabajo [65, 66, 77, 78].
- Iteraciones de duración fija (*timebox*) [62, 66, 77, 78].

En general, se mantiene la correspondencia con lo planteado al inicio del presente capítulo como factores decisivos a la hora de implementar un enfoque de gestión ágil para proyectos de ciencia de datos. Se listan algunos aspectos relevantes a considerar:

- La planificación del proyecto, no en el sentido predictivo de una metodología de tipo cascada, sino para favorecer la entrega de valor en función de la priorización que determine el cliente / usuario interesado en el proyecto [63, 64, 65]. La sesión de planificación de una iteración presenta utilidad para que el equipo de trabajo pueda estimar tiempos y esfuerzos a aplicar en cada tarea; y determinar qué entregables se espera obtener al finalizar la iteración [77, 78]. Por el lado de la planificación de versiones del producto, aunque abierta a cambios, permitirá conocer al negocio qué esperar y en qué tiempos, al mismo tiempo que servirá para estimar los costos derivados de la ejecución del proyecto no solo a nivel de recursos humanos sino también a nivel de la infraestructura requerida para la puesta en producción de la solución [50, 52, 62, 77].
- Como se ha mencionado previamente, las técnicas o prácticas de gestión que un equipo podría aplicar para un proyecto son adaptables [61, 62]. De un proyecto a

otro o entre diferentes equipos, un mismo marco general de trabajo podría variar en pos de mejorar el ambiente de trabajo y la calidad de los resultados que son obtenidos iteración a iteración. En este contexto, las reuniones de retrospectiva son el escenario ideal para este tipo de planteos y modificaciones que permitirán optimizar el trabajo del equipo [64, 65, 77].

- La retoalimentación del usuario / cliente del producto en desarrollo también se presenta como un aspecto fundamental del desarrollo, para lo cual las reuniones de demo o presentación del incremento implementado en cada iteración tienen gran importancia [65, 66]. En tales espacios, junto a las sesiones de planificación de cada iteración se puede obtener información de relevancia para el equipo a fin de asegurar que el desarrollo en ejecución efectivamente aporte valor a la organización a medida que avanza [62, 76]. En estas reuniones la visualización de avances, aunque sean datos generados o resultados concretos de investigación pendientes de implementación, permitirá tener una visión concreta del avance logrado y encontrar oportunidades de mejora para la calidad del producto tanto a nivel de funcionalidades como a nivel de arquitectura o infraestructura [50, 51, 77].

- Hasta el momento se ha omitido mencionar cuáles son los items de trabajo a emplear, esto se debe a que existen posturas a favor de utilizar historias de usuario [63, 66], otros a favor de preguntas-problema [79] y otros con un planteo mixto entre tareas denominadas lineales (más orientadas a desarrollo) y circulares (más orientadas a investigación - optimización) [80]. En todos los casos, para los fines de este trabajo, se considera que el mismo equipo deberá determinar qué formato de captura de requisitos desea emplear en función de su contexto particular. Sin embargo, algo en lo que existe mayor consenso es en la definición de qué entregables se generarán con la implementación de un item de trabajo. Esto permite tener previsibilidad y no generar falsas expectativas en torno al trabajo del equipo en una iteración en particular [65, 66, 78].

- Finalmente, las iteraciones se presentan como un concepto a emplear bajo la consideración de que sean de duración fija [66, 77, 78]. Por un lado, se busca que el desarrollo de la PoC o el MVP del proyecto se limiten a una serie de iteraciones estimadas previamente. Esto a fin de poder descartar el avance en caso de que se considere adecuado y no proseguir con un proyecto que no agregará valor a la organización ni solucionará problemas asociados a su negocio [66, 77]. Por otra

parte, una vez que el producto objetivo se encuentre en una instancia de producción, las iteraciones servirán también para delimitar los tiempos de las acciones de investigación - optimización a fin de que no se extiendan más de la cuenta y se conviertan en un desperdicio de recursos [61, 76, 81].

Con esta recopilación y la realizada en la sección anterior, que abarcó solo a prácticas ágiles, se dispuso de una base sobre la cual se pasó a definir la solución propuesta al problema del presente trabajo.

4.4 Elaboración de la propuesta de solución

Con los resultados del trabajo descripto en las secciones previas se contó con un panorama de las técnicas o prácticas ágiles que son más utilizadas en la industria. De manera similar, se realizó un acercamiento a las características principales de las alternativas para aplicar un enfoque de gestión ágil para proyectos de ciencia de datos. En ambos casos las técnicas y/o prácticas fueron diferenciadas en tanto se tratasen de una cuestión relacionada estrictamente con la gestión del flujo de tareas del proyecto en cuestión o de tareas de índole más técnica o de ingeniería.

Tal como se planteó en la definición del problema del presente documento, el objetivo es generar una propuesta de marco de trabajo para aplicar técnicas ágiles en la gestión de un proyecto de ciencia de datos acotando su caracterización para pequeñas y medianas organizaciones. Es por esto que necesariamente, se comenzó por acotar a qué organizaciones se las consideraría pequeñas o medianas.

4.4.1 Pequeñas y medianas organizaciones

En términos generales, en Argentina para que una organización pueda ser considerada como pequeña o mediana debe cumplir con ciertos requisitos que son definidos por la Administración Federal de Ingresos Públicos (AFIP). De esta manera, una Pequeña y Mediana Empresa (PyME) será aquella que no se exceda en un monto determinado de facturación en sus últimos tres ejercicios contables y no supere una cantidad específica

Categoría	Cantidad máxima de empleados	Límite de facturación anual
Micro	9	AR$ 9.9 M
Pequeña	30	AR$ 59.7 M
Mediana - Tramo 1	165	AR$ 494.2 M
Mediana - Tramo 2	535	AR$ 705.8 M

Tabla 4.3: Características de PyMEs según AFIP.
Fuente: elaboración propia basado en [82].

Categoría	Cantidad de empleados
Micro	hasta 9
Pequeña	entre 10 y 49
Mediana	entre 50 y 200
Grande	más de 200

Tabla 4.4: Características de PyMEs según el OPSSI de la CESSI.
Fuente: elaboración propia basado en [83].

de empleados registrados. Para ver los detalles de la clasificación determinada por la AFIP [82] se presenta la tabla 4.3. Es necesario mencionar, que tanto en la tabla presentada como en la clasificación en general las micro empresas también forman parte de la categoría PyME a fines prácticos, el mismo criterio fue tomado en el resto del documento.

A partir de los datos de la AFIP se pudo contar con una clasificación general para las organizaciones objetivo. Sin embargo, estos valores, específicamente aquellos referidos a la cantidad de personal contratado fueron revisados en detalle sobre el área de las organizaciones que brindan servicios relacionados con las tecnologías de la información. Esto se debió a que existía la posibilidad de que en ese subconjunto de empresas las condiciones fueran diferentes. Es así como se recurrió a los datos de la Cámara de Empresas de Software y Servicios Informáticos (CESSI) que anualmente reporta diferentes estadísticas a través del Observatorio Permanente de la Industria de Software y Servicios Informáticos (OPSSI). En el reporte publicado en el año 2019 (con datos obtenidos durante 2018) la caracterización de las empresas varía con respecto a lo establecido por la AFIP [83]. Los datos en cuestión se presentan en la tabla 4.4.

Como se puede observar al comparar ambas tablas, las empresas relacionadas con las tecnologías de la información manejan valores diferentes en torno a la cantidad de personal a partir del cual pasan a considerarse medianas y grandes. Sin embargo,

para los fines del presente trabajo, las categorías de micro y pequeña empresa se consideran como una representación adecuada de la cantidad de personal disponible en una organización / empresa para la ejecución de un proyecto de ciencia de datos. Esto se debe a que, justamente, las empresas de las categoría micro y pequeñas conforman la amplia mayoría según los datos que publica el Ministerio de Trabajo de la Nación (Argentina) a través del Observatorio de Empleo y Dinámica Empresarial (OEDE) con base en los datos del Sistema Integrado Previsional Argentino (SIPA). En tales estadísticas, cuya última actualización se remonta al año 2018, sobre el total de 5406 empresas del área de informática, un 93% se corresponde a pequeñas y micro empresas [84].

En cuanto a la conformación en particular de un equipo de trabajo para un proyecto de ciencia de datos, las recomendaciones de la industria y academia se basan en los siguientes criterios al hablar de organizaciones de tamaño pequeño a mediano [85, 86, 87, 88]:

- Una persona con conocimiento del dominio del negocio a partir del cual surge el o los problemas a resolver en el proyecto
- Una persona (al menos) con conocimiento de infraestructura y/o de programación para generación y gestión de los flujos de trabajo
- Una persona (al menos) con conocimiento en ciencia de datos, capaz de realizar las diferentes tareas involucradas en el desarrollo del producto de datos en cuestión

Evidentemente a medida que la organización destine más personal a este tipo de proyectos, los roles podrán variar al igual que la estructura del equipo [87]. En tal escenario, la división de responsabilidades del último rol entre un perfil científico y uno de ingeniería sería un primer paso. Para el caso de infraestructura y arquitectura, dos personas podrían, con diferentes conocimientos, abordar cada una de esas áreas para poder avanzar en la escalabilidad de las soluciones en desarrollo. Por otra parte, además del rol de conocimiento del negocio, una persona con experiencia en el análisis de los datos del dominio podría ser de gran utilidad para garantizar la calidad de los productos en desarrollo al momento de ampliar un equipo [85, 88].

En este apartado, se utilizaron datos correspondientes a empresas, organizaciones con fines lucrativos, sin embargo el término "organizaciones" empleado en la

definición de los objetivos del presente trabajo hace referencia tanto a esas empresas como a organizaciones cuyo objetivo final no sea lucrativo en sí mismo pero posean la infraestructura tecnológica y el personal como para ejecutar proyectos de ciencia de datos. En la continuidad del trabajo, ambos términos: empresas y organizaciones, serán empleados indistintamente con una clara preferencia por el segundo debido a su generalidad.

4.4.2 Definición de la propuesta de gestión ágil para proyectos de ciencia de datos

Con las características y perfiles mencionados previamente se logró caracterizar tanto a las organizaciones como al equipo de trabajo base para la propuesta del presente trabajo. Esta estructura y sus limitaciones (principalmente a nivel de personal) fueron consideradas a fin de que la misma no resultara abrumadora y incremente la burocracia de procesos o flujos de trabajo que perderían el *status* de ágiles y dificultarían el crecimiento de una iniciativa de ciencia de datos en una organización del tipo objetivo.

El marco de trabajo consta de una serie etapas y elementos a generar por cada una de ellas. En todo momento se aplican diferentes técnicas de gestión y en la construcción de los items mencionados intervienen o se ven representadas diferentes tareas de ciencia de datos relacionadas a una o más fases de las metodologías relevadas. El objetivo perseguido en su organización ha sido el de mantener los criterios básicos de integración de la gestión ágil a este tipo de proyectos, es por eso que se aplicaron los siguientes criterios:

- Se trabaja a través de iteraciones.
- Los productos resultantes son frecuentemente presentados y puestos a disposición de los usuarios finales / clientes.
- Se cuenta con diferentes instancias de revisión de tales resultados a fin de obtener retroalimentación constante que permita tanto adaptarse a los cambios como determinar la viabilidad del proyecto.
- Los requisitos son definidos de manera general a través de historias de usuario, salvo que el equipo determine una estrategia alternativa.

- Los incrementos del producto a generar en cada iteración no necesariamente son un entregable sino que podrían ser resultados de fases intermedias del proceso de ciencia de datos, por ejemplo: un *dataset* resultante de tareas de preparación de los datos.

Considerando lo mencionado previamente, se presenta la progresión de etapas de la propuesta:

1. Se comienza con una fase de **Inicio** donde se establecen lineamientos generales, objetivos y ciertos parámetros de ejecución. Al final de esta fase se contará con una planificación de alto nivel del proyecto a partir de la cual se podrá estimar en cuántas iteraciones se desarrollará la PoC del mismo. Esto permitirá identificar un primer punto de control y evaluación a fin de determinar la viabilidad de la iniciativa.

2. Luego, se pasa a realizar una **Iteración cero** donde se realizan tareas técnicas relativas a la configuración básica de las herramientas y entornos de trabajo a utilizar.

3. En este punto se cuenta con un ambiente operativo definido y se puede comenzar a ejecutar una serie de ***N* iteraciones** de trabajo donde se buscará desarrollar la PoC de la solución al problema o necesidad de origen. La cantidad de iteraciones se habrá definido a partir de la planificación mencionada como salida de la fase de inicio. Con este primer resultado se realizará una revisión entre todos los interesados y se determinará la viabilidad del proyecto. Dependiendo de las características del problema abordado y las condiciones de desarrollo puntales de la PoC podría realizarse el **despliegue** de la misma. En tal caso se ejecutarían, si el equipo lo encuentre adecuado, las acciones de la fase del mismo nombre en una nueva iteración.

4. En caso de haber **aprobado la PoC**, el siguiente paso será avanzar en el desarrollo del MVP del proyecto. En este sentido podría ser necesario actualizar algunos aspectos de la planificación de versiones dado que se contará naturalmente con más conocimiento del contexto que al momento de las estimaciones iniciales. A partir de este posible ajuste se estará en condiciones de determinar el número de iteraciones que serán necesarias para la implementación del MVP y su posterior

despliegue. Un **rechazo de la PoC** podría significar tanto la cancelación del proyecto como un cambio en sus objetivos y una nueva planificación inicial volviendo al paso previo.

5. El paso siguiente es la ejecución de ***N* iteraciones** de trabajo donde se abordará la implementación del MVP del proyecto. Al cierre de cada una de ellas se presentarán los avances logrados y, en caso de corresponder, se sumarán al despliegue que haya sido realizado de la PoC a la espera de esa primera versión. En tales instancias, final de cada iteración, se tendrá la posibilidad de realizar adaptaciones o cambios que el negocio considere necesarios para que los resultados obtenidos y a obtener aporten valor a la organización.

6. Una vez finalizado el MVP se deberán realizar los pasos de **despliegue** adecuados. Debido a su complejidad, podría definirse una iteración con este objetivo a fin de registrar adecuadamente el trabajo requerido a tal efecto.

7. Estando en producción el MVP será momento de ajustar la **planificación de versiones** y determinar los siguientes pasos del proyecto en términos de agregado de funcionalidades, mantenimiento y seguimiento del producto que será utilizado por los usuarios finales. De esta manera se podrá conocer de manera fehaciente cuáles serán las características de la próxima versión a implementar y cuándo se estima que estará disponible. En este punto se vuelve a iniciar el ciclo de: iteraciones de trabajo, obtención de una nueva versión del producto y despliegue.

De esta manera queda conformado el planteo general de la propuesta generada, la distribución de etapas se puede observar en la figura 4.1, en las próximas secciones del documento se describirán en forma detallada cada una de las fases mencionadas.

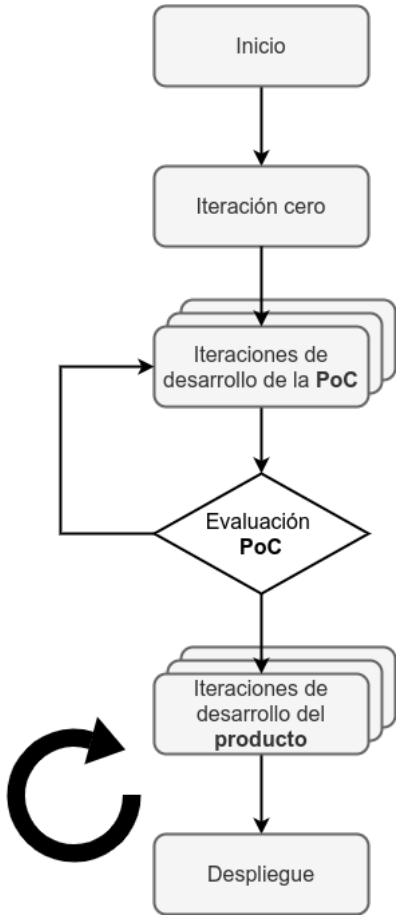

Figura 4.1: Fases de la propuesta desarrollada.
Fuente: Elaboración propia

4.4.2.1 Inicio del proyecto

En cuanto al inicio del proyecto, antes de comenzar con el trabajo en sí, en la tabla 4.5 se presentan los elementos a generar, sus características y las tareas involucradas en su creación.

Artefacto a generar: *Product backlog*	
Prácticas / tareas de gestión ágil	- Historias de usuario - Estimación de complejidad - Priorización de requerimientos
Tareas de ciencia de datos	- Definición de objetivos del proyecto - Determinar producto de datos - Identificar orígenes de datos - Selección de métricas de evaluación - Definición de umbrales de éxito
Artefacto a generar: Planificación de versiones	
Prácticas / tareas de gestión ágil	- Velocidad del equipo (estimar) - Duración de cada iteración (establecer)
Tareas de ciencia de datos	- Planificación inicial del proyecto
Artefacto a generar: Organización del trabajo	
Prácticas / tareas de gestión ágil	- Determinar reuniones a emplear (sobre la base de Scrum) - Determinar *timebox* para las reuniones - Establecer definición de listo - Establecer definición de hecho - Determinar instrumentos de interacción con usuario / cliente (medios, tiempos)
Tareas de ciencia de datos	–

Tabla 4.5: Fase de inicio del proyecto.
Fuente: elaboración propia.

En esta etapa inicial se espera clarificar tanto los objetivos del proyecto, como el tipo de producto de datos a implementar. Esto se logra a través de cada historia de usuario donde se especifica la funcionalidad o resultado deseado por el cliente junto a una estimación de su complejidad (relativa a las demás historias de usuario detectadas en ese momento), su valor para el negocio (que establecerá su priorización en cuanto al desarrollo), el/los orígen/es de datos para su implementación (si fuera posible establecerlos en ese momento) y los criterios de aceptación para validar que su desarrollo fuera finalizado. Todas las historias de usuario relevadas en esta instancia conforman

el *backlog* del producto a desarrollar. Generalmente, las historias serán escritas en forma conjunta entre los *stakeholders* y el equipo de trabajo, los primeros expondrán la necesidad a cubrir, el equipo la interpretará y entre ambos llegarán a un acuerdo al respecto de su definición. De manera similar, la complejidad de una historia de usuario se estima en forma conjunta entre todos los involucrados en esta instancia. Se parte de un acuerdo inicial en torno a qué métrica se va a emplear; la tecnología a utilizar para su implementación; y qué tareas se incluyen, análisis, diseño, implementación, pruebas y/o despliegue. Sobre esta base, se otorga un valor en la métrica seleccionada a una historia de usuario y posteriormente se la compara con las demás a fin de estimarlas en función de que sean más o menos complejas que la primera.

Establecer los indicadores de rendimiento a medir para la solución y los umbrales de éxito a considerar también será de ayuda para que, junto a la planificación inicial del proyecto, se pueda tener una noción del momento y la forma en el que la PoC del producto será evaluada y se podrá determinar la viabilidad de la iniciativa. Finalmente, se incluyen algunas actividades de coordinación general que obedecen a la organización general del equipo en torno a cuestiones del día a día que es conveniente establecer para su correcta gestión.

Los artefactos propuestos en la tabla son suceptibles de ajustes y adaptaciones que el equipo de trabajo y/o el cliente / usuario decidan que serán más aplicables en función de factores como: su forma habitual de trabajo, restricciones aplicables al caso, entre otros. En general, con estas acciones se estarían ejecutando las tareas correspondientes a la fase de comprensión del negocio de las metodologías de ciencia de datos antes mencionadas (CRISP-DM y TDSP por ejemplo).

4.4.2.2 Iteración cero

En una segunda etapa, se propone realizar una primera iteración especial en la que se realice la configuración general de los entornos y herramientas software de soporte que serán empleadas durante la ejecución del proyecto. Esto se debe a que son tareas íntegramente técnicas y que es conveniente realizar en una iteración individual [89] a fin de que no afecten la implementación de alguna historia de usuario una vez iniciado el proyecto. Las principales tareas de esta instancia se pueden observar en la figura 4.2 y se detallan a continuación:

- **Configuración de entornos**: se realiza la configuración inicial del entorno a emplear para desarrollo, pruebas y despliegue. En una instancia temprana del proyecto puede que solo involucre la instalación de librerías y/o herramientas para acceso y procesamiento de datos en función de los objetivos definidos en la fase previa. Es de utilidad no solo para la configuración de un servidor o una instancia que cumpla esta función sino también para los entornos de trabajo de todos los involucrados con las diferentes herramientas a utilizar. Inclusive se podría tratar de generar las cuentas o credenciales necesarias para el acceso a determinados servicios que fueran a ser necesarios o guarden relación con los objetivos planteados.
- **Configuración de la estructura del proyecto**: es recomendable seguir una estructura para la organización interna de los archivos que conforman a la solución del problema a resolver (código, documentación, referencias, entre otros) para que se simplifique su acceso y la misma no sea dependiente de una única persona. Se trata de una práctica bastante habitual a raíz del uso de diferentes *frameworks* en desarrollo de software donde al iniciar un nuevo trabajo se crea automáticamente la estructura de directorios para ordenar los archivos [90, 91].
- **Especificar los estándares de codificación**: esta es una práctica ligada a la anterior ya que se complementan. El establecimiento de un estándar de codificación o documentación puede ser de utilidad para la definición de nombres, estrategias de importación de librerías, gestión de dependencias, manejo del sistema de control de versiones, entre otras acciones propias del desarrollo que es conveniente acordar entre los integrantes de un equipo antes de iniciar su ejecución para facilitar el trabajo [90].
- **Configuración del repositorio de versionado**: otra buena práctica referida a la gestión de todos los archivos del proyecto es generar y utilizar un método de versionado del producto. No se limita solo a archivos de código fuente, sino que otros tipos de archivos se pueden versionar también logrando resguardar ante cambios imprevistos el avance logrado por el equipo [92].
- **Definición de arquitectura de la solución**: en esta instancia se podría definir una arquitectura básica de la solución considerando los objetivos definidos en la fase de inicio y asociando a los mismos un tipo de producto en particular a generar para su solución. Esta arquitectura, naturalmente, deberá ser modificable para adaptarse a los cambios que puedan sucederse más adelante.

- **Habilitación de la herramienta de seguimiento de incidencias**: se trata de la configuración y habilitación para el equipo de la herramienta en la cual se realizará el seguimiento de las incidencias o tareas del proyecto por parte de todos sus integrantes. En la configuración se podrían incluir aspectos tales como: accesos, permisos, identificación de la iniciativa, estructura básica, política de notificaciones, modelo de trabajo, duración de las iteraciones, formato de las incidencias, entre otros.

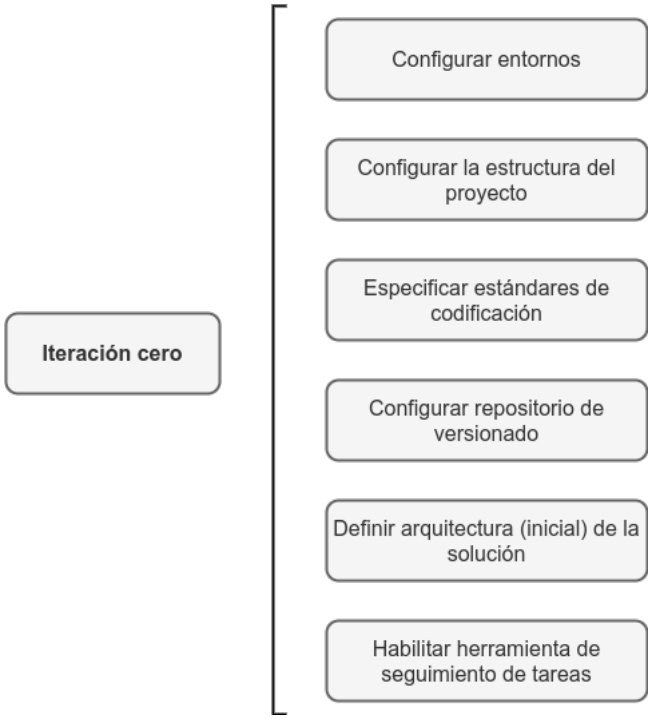

Figura 4.2: Detalle de la iteración cero.
Fuente: Elaboración propia

Estas tareas se relacionan con las fases de preparación del proyecto de las metodologías listadas en las secciones previas, como ser: ASUM-DM, TDSP, Agile KDD, entre otras. La recomendación de integrar este tipo de tareas busca que el mismo sea documentado de forma simple y que tanto su seguimiento como su implementación, siguiendo las buenas prácticas aplicables, no sean actividades que dificulten el avance en su desarrollo. En cada organización, estas tareas podrán variar en función de lineamientos previos que pudieran existir y la adaptación de estas tareas al contexto de cada equipo deberá ser realizada.

4.4.2.3 Inicio de una iteración

El punto de partida de una iteración será la planificación de los items a implementar en la misma, esta es una de las prácticas de gestión que se reconocen como centrales para una iniciativa ágil dado que se establecen los objetivos de la iteración y se determinan los incrementos del producto a desarrollar en su ejecución [63, 64, 65, 66, 73, 77]. Las tareas de esta etapa, visibles en la figura 4.3, son:

- **Refinamiento del *backlog* del producto**: sobre el listado de historias de usuario se podrían realizar modificaciones, tanto de su estimación inicial como de su contenido o prioridad en función de las necesidades del negocio. Esta tarea, también apunta a agregar detalles en los items de trabajo que se vayan a incorporar a la próxima iteración [71]. Esto se da en el siguiente contexto: una historia de usuario podría estar vagamente definida en el *backlog* del producto, sirviendo como un recordatorio de una funcionalidad o requerimiento a implementar, estimada en forma inicial y sin mayores detalles. Antes de proceder con su implementación en una iteración, se debe alcanzar el estado que haya sido definido por el equipo, cliente / usuario incluido, a través de la definición de listo (*Definition of Ready* - DoR por su sigla en inglés) en la que se establecieron los componentes mínimos con los que debe contar una historia de usuario antes de ser pasada a desarrollo en una iteración. En la DoR se pueden especificar cuestiones tales como: necesidad de criterios de aceptación, presencia de ejemplos de resultados esperados, responsables de su aprobación o prueba, entre otros items que serán acordados entre los miembros del equipo. La realización de estas tareas es una acción integrada dentro de esta etapa para que al iniciar la planificación de una iteración las historias de usuario involucradas se encuentren listas para su desarrollo, aparte de brindar la oportunidad al cliente / usuario de modificar las prioridades de avance en función de oportunidades de negocio o cuestiones similares que pudieran haber surgido desde la última iteración.

- **Selección de las historias de usuario para el *backlog* de la iteración**: en este caso se trata de tomar aquellos items que van a ser desarrollados en la iteración próxima a comenzar y llegar a un punto de entendimiento de los mismos por todos los miembros del equipo. Esto permitirá minimizar errores de interpretación en

el desarrollo, además de servir para realizar dos acciones relacionadas con cada historia de usuario como son:

  - **Descomposición en tareas técnicas**: una historia de usuario no tendrá una descripción técnica detallada en el común de los casos, es por ello que antes de proceder con su inclusión en una iteración se debería descomponerla en las tareas de índole técnico necesarias para su implementación. El nivel de granularidad y la especificidad de las tareas podrá variar de un equipo a otro en función de sus preferencias con respecto al seguimiento de las mismas, inclusive se podría utilizar alguna plantilla como la propuesta en el marco de trabajo TDSP [93]. Esta descomposición también se considera de utilidad para poder realizar la estimación de tiempos de implementación de cada item de trabajo.
  - **Estimación detallada**: a fin de establecer qué historias de usuario se podrán o no incluir en la iteración, se debe refinar su estimación pasando de marcar complejidad a tiempos de implementación en la unidad que desee emplear el equipo. En relación al punto previo, el listado de tareas a ejecutar para su desarrollo será de utilidad para establecer esto, pudiendo servir también para determinar que una historia podría requerir ser dividida en dos o más a fin de simplificar su resolución en los tiempos de una iteración. Finalmente, a modo de recomendación de técnica a emplear para estas acciones, se propone un método basado en opiniones como podría ser *planning poker* [71]. Esta tarea la ejecuta el equipo de trabajo con base en las historias procesadas en el paso previo.

- **Armado del tablero de la iteración**: una vez seleccionadas las historias de usuario a implementar y sus correspondientes tareas será momento de plasmar la pila de trabajo de la iteración en un tablero del estilo Kanban [71]. La cantidad y objetivo de cada columna podrá ser definida por cada equipo, pudiendo ir desde un enfoque "tradicional" de tres columnas: pendiente, en proceso, finalizado; hasta otros que integran columnas específicas para instancias de prueba, documentación o integración. En este sentido, el uso de una herramienta software podría ser de gran utilidad para brindar soporte a esta tarea y al seguimiento que implica su utilización.

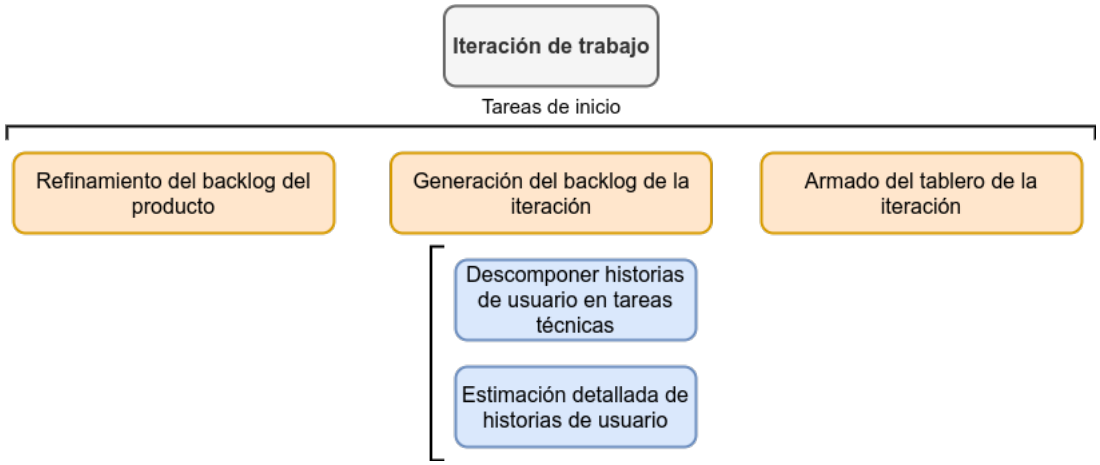

Figura 4.3: Tareas de inicio de cada iteración.
Fuente: Elaboración propia

Como se puede observar, esta etapa no guarda relación estricta con una de tipo técnica en las metodologías o modelos de proceso para proyectos de ciencia de datos. Sin embargo, en algunas de ellas se encuentran planteos con respecto a la gestión de las tareas involucradas [14, 15, 16, 50] y esta etapa se considera como un enfoque iterativo de tales acciones. La planificación general del proyecto no será generada en una única vez sino que será construida iteración a iteración aunque con una visión que se establece inicialmente con la planificación de versiones. En todos los casos, se remarca la característica de adaptabilidad y flexibilidad ante cambios que debe tener cualquier modelo de procesos para este tipo de proyectos. Esto se debe, a la incertidumbre y tendencia de requerimientos volátiles que usualmente tienen las iniciativas de ciencia de datos tal como se ha mencionado previamente [61, 65, 66].

4.4.2.4 Desarrollo de una iteración

Con cada iteración en marcha, las tareas previstas en el apartado de gestión se reducen, mientras que se integran tareas técnicas de ciencia de datos a fin de avanzar sobre la implementación del producto en desarrollo. Por el lado de las prácticas de gestión aplicables se pueden mencionar:

- **Reunión del equipo**: si bien en su origen se plantea como diaria, dado que un equipo podría no estar afectado totalmente al proyecto podría no ser aplicable. En

este sentido, se recomienda su utilización para lograr la comunicación necesaria entre los integrantes del equipo, con la frecuencia que determinen y represente un beneficio para el desarrollo del proyecto. Esta reunión suele realizarse en forma breve y se busca responder tres preguntas: ¿Qué se hizo en el día previo? ¿Qué se va a hacer en el día en curso? ¿Qué problemas se han detectado?

- **Programación de a pares**: ante tareas cuya complejidad así lo justifique o problemas que se puedan presentar, los integrantes del equipo podrían trabajar en forma conjunta sobre una misma estación de trabajo en un problema particular a fin de abordar su solución de manera colaborativa. Una vez superada la dificultad, cada uno volvería a su esquema de trabajo habitual. Como resultado de esta interacción podría documentarse la solución generada para servir como recurso de información para futuras ocurrencias de problemas similares.

- **Pruebas unitarias**: dependiendo del tipo de producto de datos en desarrollo, se podrían implementar conjuntos de pruebas unitarias para verificar su correcto funcionamiento. En este caso, las pruebas son codificadas por la misma persona que realiza la implementación y pueden servir para aplicar los criterios de aceptación de la historia de usuario que se encuentre en desarrollo.

- **Seguimiento de lineamientos**: durante la ejecución de la iteración cero, el equipo debería haber acordado sobre los estándares de codificación a emplear, el modelo de estructura de directorios para el proyecto, diferentes estrategias a seguir en aspectos de gestión del producto. Entre ellos se pueden mencionar: estrategia de integración, automatizaciones, gestión de entornos e inclusive en cuestiones del trabajo diario, por ejemplo: en qué casos se considera una alternativa la programación de a pares. Estas cuestiones, junto a otras como las DoD o DoR que se relacionan de forma más estrecha con la gestión, deberán ser respetadas durante las iteraciones. Además, en caso de desear proponer un cambio sobre alguno de estos aspectos, el espacio será el de la reunión de retrospectiva al final de la iteración.

Como se pudo observar en el listado previo, en general, se trata de cuestiones en las que se aplican los criterios y acuerdos planteados por el equipo junto al cliente en la etapa de inicio del proyecto. En lo que respecta a las tareas de índole técnico, durante la ejecución de la iteración se pueden incluir algunas o todas de las siguientes tareas:

- **Análisis exploratorio de datos**: para poder determinar si los datos disponibles pueden ser empleados efectivamente para resolver el problema objetivo del proyecto o la iteración. Si bien este tipo de análisis, posiblemente, será más común en las instancias iniciales de trabajo, a medida que se presenten alguna inquietudes sobre las características de los datos podría ser de utilidad volver a realizarlo.

- **Análisis de la calidad de los datos**: los problemas de calidad de datos frecuentemente forman parte de las iniciativas de ciencia de datos [62, 65]. Antes de proceder con el desarrollo de algún producto podría ser de gran utilidad analizar la calidad de los *datasets* disponibles para identificar problemas que pudieran afectar el avance del proyecto y buscar alternativas de solución para los mismos.

- **Ingeniería de atributos**[3]: el diseño y la implementación de nuevos atributos para el *dataset* sobre el cual se esté trabajando, constituye una actividad que puede servir para integrar nuevas variables al análisis o en los modelos a generar en función de los objetivos planteados.

- **Definir un circuito de actualización de datos**: en caso de corresponder, los datos a emplear por el producto deberán ser actualizados con cierta frecuencia. La definición de los pasos a seguir para esta actividad, junto a la posible implementación de un proceso automatizado son acciones que forman parte del desarrollo de la solución y su mantenimiento una vez que se encuentre en funcionamiento.

- **Reducción de dimensionalidad del *dataset***: la cantidad de filas y columnas del *dataset* afecta al rendimiento de los métodos de procesamiento a emplear. Además, podría pasar que un conjunto de filas con datos incorrectos deba ser filtrado del conjunto de datos para no obtener resultados erróneos a partir de su procesamiento o entrenamiento de un modelo. De igual manera, las columnas que no aporten información para el problema a resolver o cuyos datos sean considerados como no relevantes, deberían ser eliminadas.

- **Segmentación del *dataset***: en el proceso de construcción de un modelo, de predicción o clasificación por ejemplo, es necesario contar con un *dataset* de entrenamiento para la técnica a utilizar y uno de prueba a fin de evaluar el modelo obtenido. Esta operación, aunque incluída en algunas herramientas disponibles, se debe realizar con ciertos recaudos para garantizar una correcta generación de

[3]Se toma la traducción literal del vocablo *feature engineering*.

ambos conjuntos de datos que respete una distribución de clases adecuada. Inclusive se consideran aplicables métodos de validación cruzada o similares que permiten reducir los problemas asociados a estos métodos de trabajo.

- **Selección y prueba de efectividad de las técnicas para modelado**: la selección de técnicas a aplicar deberá estar condicionada por los datos disponibles y sus características, el tipo de problema que se pretende resolver y el tipo de producto que se encuentra en desarrollo. Esta tarea, que deberá ser consensuada por los integrantes del equipo, posteriormente requiere que sean ejecutadas las técnicas seleccionadas y evaluada la efectividad de los modelos generados a través de ellas. En algunos casos, serán aplicables técnicas adicionales como las de hiper parametrización o ensamblado de modelos, aunque se podría tratar de tareas a realizar una vez que se valide que la/s técnica/s seleccionada/s cumplen con criterios básicos de efectividad.

- **Definición de un plan de pruebas**: inicialmente se trata de las pruebas unitarias que cada integrante del equipo pudiera implementar sobre una o más actividades del proceso como las descriptas en estas líneas. Adicionalmente, se podría incluir un esquema de pruebas similares a las de integración en proyectos de desarrollo de software a fin de verificar cómo trabajan en forma conjunta los diferentes productos que sean generados en cada iteración. Estas pruebas podrían automatizarse mediante herramientas de integración contínua [95, 97] y generar reportes de éxito/fracaso ante ciertas acciones, los detalles de su utilización deberían ser establecidos en la iteración cero y actualizados conforme se produzcan cambios.

- **Evaluación de los resultados**: si bien el espacio para la evaluación de los resultados obtenidos será al final de cada iteración, el equipo naturalmente ejecutará en forma constante evaluaciones sobre los avances logrados considerando diferentes métricas y umbrales de éxito que hubieran sido definidos previamente. Esto permitirá identificar en forma temprana los modelos o sub-productos que serán aptos para su pase a instancias posteriores del proyecto. Además, para validar que las selecciones de parámetros, en caso de corresponder, hayan sido las adecuadas.

En este caso, las tareas de una iteración en el aspecto técnico guardan relación con las fases generales de preparación de los datos y modelado, aunque también se

pueden incluir elementos de comprensión de los datos [15, 52, 75]. El esquema general de las tareas del desarrollo de una iteración se observa en la figura 4.4.

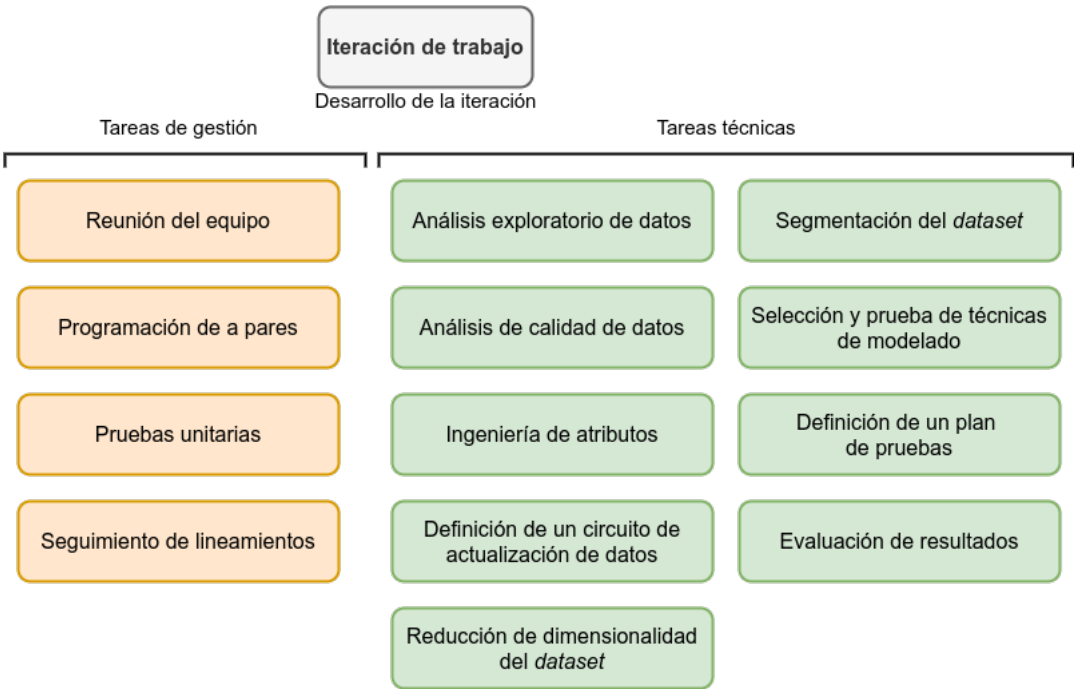

Figura 4.4: Tareas ejecutables en el desarrollo de cada iteración.
Fuente: Elaboración propia

4.4.2.5 Cierre de una iteración

Al finalizar una iteración se presenta un entregable, si bien qué será y qué no será considerado un entregable podría variar en función de las preferencias del equipo, el objetivo de esta etapa es el mismo: validar y verificar lo generado con los interesados en el proyecto. En la figura 4.5 se observan los dos eventos a realizar en el cierre de cada iteración.

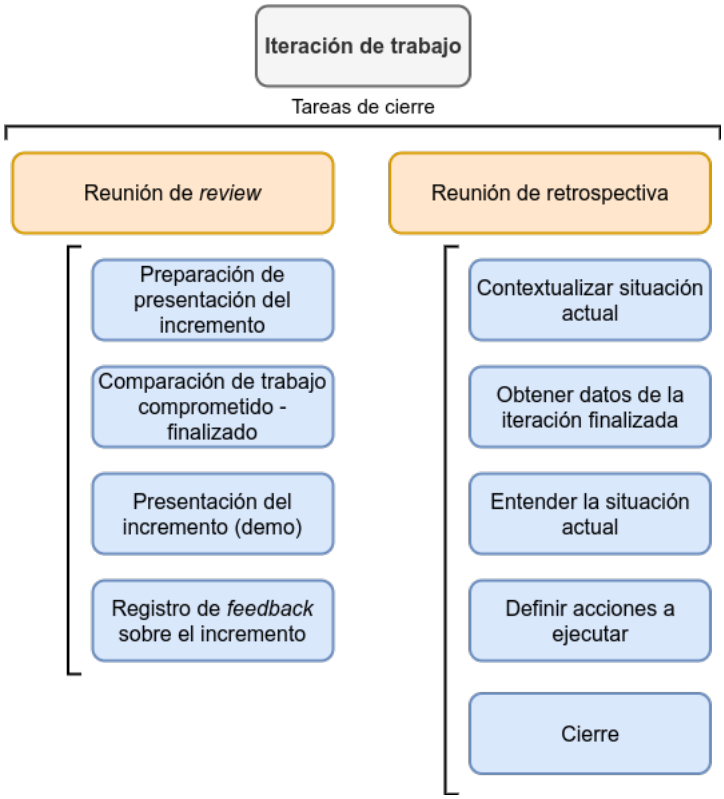

Figura 4.5: Tareas de cierre de cada iteración.
Fuente: Elaboración propia

En primer lugar se llevará a cabo una reunión de revisión, o *review*, junto a los representantes del cliente. Entre las acciones a ejecutar tanto en forma previa como durante la reunión se pueden mencionar [94]:

- Preparación del producto en el ambiente correspondiente, usualmente despliegue, mediante los procesos de integración y/o despliegue continuo. En caso de no tratarse de un producto que así lo requiera será cuestión de asegurar su presentación de manera que sea apreciable por todos los involucrados y se comprenda el valor que aporta para el proyecto y el negocio en forma general.
- Durante la reunión se podrá hacer una comprobación entre los items del *backlog* de la iteración que fueran comprometidos y los que efectivamente hayan sido implementados. En caso de haber diferencias el equipo podrá exponer las razones de tal situación y determinar, en caso de corresponder, de qué manera podrían subsanarse para evaluar su desarrollo en la siguiente iteración.

- Presentación del producto a los interesados. En particular se indicará el beneficio que aporta su desarrollo o avance para el logro de los objetivos generales del proyecto, además de mostrar su funcionamiento básico, en caso de corresponder.
- Ante presentaciones en las que algunos de los presentes brinden su opinión o *feedback* sobre el producto que se esté presentando, será tarea del equipo registrar esta retroalimentación, sea para realizar ajustes o para considerar tales observaciones en las próximas iteraciones del proyecto.
- Actualización, si fuera necesario, de los tableros de seguimiento de tareas en las herramientas correspondientes.

En esta reunión es donde se interpretan los resultados que presenta el equipo desde la visión de los usuarios/clientes y se da lugar a la colaboración entre todos los actores involucrados para determinar si las soluciones presentadas constituyen un aporte para resolver el problema de origen del proyecto en cuestión. Esta sesión será de utilidad para identificar y determinar mejoras que podrían realizarse sobre el producto en desarrollo o incluso optar por un cambio de estrategia o línea de trabajo en caso de no obtener los resultados esperados.

Posterior a la reunión de revisión, también denominada de *demo*, se podrá realizar bajo la frecuencia que se considere adecuada para el proyecto una reunión de retrospectiva [94]. La idea principal de dicha reunión es identificar mejoras aplicables no sobre el producto en sí mismo, sino sobre el modo en el que se está ejecutando y gestionando el proyecto. Es una instancia importante del proceso ya que es la base sobre la cual se podrá lograr la mejora continua de la iniciativa. Se establece una serie de acciones a realizar en esta reunión:

1. **Contextualizar**: se debe poner en situación a todos los participantes, usualmente el equipo de trabajo. Para esto se pueden utilizar diferentes técnicas que permitan lograr concentración y ubicación en torno a los objetivos a perseguir con esta reunión. También podría revisarse la agenda prevista para la misma con una distribución de tiempos y actividades.
2. **Obtener datos**: se trata de un momento en el que los presentes pasan a expresar en términos cuantitativos su opinión sobre lo que pudo haberse hecho bien o mal

durante la última iteración (o iteraciones en caso de una frecuencia de reuniones diferente). También será el momento para identificar el estado de las propuestas de mejora que pudieran haber surgido en una retrospectiva anterior.

3. **Entender la situación**: sobre la selección de problemas u objetivos del paso previo se buscará entender por qué se ha llegado a esa situación y expresar ideas aplicables para su resolución. Las mismas nacen del equipo de trabajo y se buscará la interacción entre los involucrados a fin de lograr una propuesta adecuada por cada problema detectado.

4. **Determinar acciones a ejecutar**: teniendo los resultados de la tarea previa se podrán definir objetivos específicos para llevar a cabo las acciones detectadas para corregir o minimizar el impacto de los problemas detectados. Estas acciones deberán ser estimadas e incluidas dentro de una o más iteraciones y serán revisadas en las próximas reuniones de retrospectiva.

5. **Cierre**: se brinda un espacio para analizar la reunión y sus resultados, identificando también qué aspectos son mejorables de cara a una próxima reunión. Esto es importante para que la misma no sea una carga para el equipo y realmente cumpla su objetivo de ser la base para la mejora continua del proyecto.

En estas reuniones se está realizando el conjunto general de operaciones involucradas en la fase de Evaluación que se describe en las metodologías o marcos de trabajo de ciencia de datos [15, 52, 73, 75]. También, particularmente la reunión de retrospectiva, tiene relación con lo que se denomina gestión del conocimiento o de las lecciones aprendidas de los proyectos [51, 72], con la salvedad de que en un enfoque ágil no serán realizadas al final del proyecto sino iterativamente dentro del mismo.

4.4.2.6 Despliegue

A través de las iteraciones del proyecto se estará desarrollando un producto que en algún momento deberá ser puesto a disposición de sus usuarios finales. El pasaje a una instancia de producción es una actividad que en el marco de trabajo propuesto se deberá realizar con frecuencia y que involucra desde el apartado de gestión actividades como las siguientes:

- Definición y ejecución de los flujos de operaciones de integración y despliegue continuo (CI[4] y CD[5], respectivamente) [95, 96] a fin de que el producto se encuentre en el ambiente adecuado para su utilización. Estas acciones se relacionan con la automatización de tareas como las relacionadas a la ejecución de pruebas unitarias y/o de integración y permitirán disminuir los tiempos de despliegue. También se incluyen en este punto las tareas relacionadas a la automatización de las actualizaciones que pueda sufrir el producto, tanto a nivel de funcionalidad como a nivel de los datos y/o modelos que utiliza para cumplir con sus objetivos.
- Realización de una o más demostraciones funcionales con los usuarios finales. A medida que el producto cambie, podría ser necesario realizar acciones de capacitación con los usuarios a fin de que comprendan cómo utilizar correctamente el recurso que se disponibiliza para ellos.
- Pruebas de integración con otras herramientas que sean utilizadas por los usuarios, en caso de corresponder. Se incluye esta actividad dado que en ciertos entornos, un producto que mantenga interacción con otras soluciones o sistemas de los usuarios deberá ser verificado y validado para evitar inconvenientes derivados de su utilización.

En lo que respecta a las operaciones de índole técnico vinculadas a esta etapa se concentran las ubicadas en las fases de Despliegue / Implementación / Implantación de las metodologías de ciencia de datos. Así como también algunas acciones vinculadas al uso de los resultados obtenidos, de cualquier tipo, descripto en tales procesos. Un listado de estas actividades se presenta a continuación [94]:

- Generación de un reporte del producto generado y desplegado.
- Entrega del acceso a los usuarios.
- Definición de la arquitectura final de la solución.
- Seguimiento y monitoreo mediante indicadores previamente seleccionados.
- Desarrollo de integraciones con otros artefactos o sistemas.

[4] *Continuous Integration* por su sigla en inglés.
[5] *Continuous Delivery / Deployment* por su sigla en inglés.

- Soporte y mantenimiento.

Como corolario de esta etapa el producto deberá estar en manos de los usuarios finales y comenzar a ser utilizado para volcar el beneficio esperado a la organización.

4.4.2.7 Situaciones a considerar

En este apartado se dará tratamiento a dos situaciones que difieren del normal flujo de desarrollo de una iteración o proyecto y que podrían ocurrir en cualquier iniciativa de ciencia de datos. Estos problemas deben ser igualmente gestionados y como parte de esa necesidad se presentan alternativas de acción a seguir:

Cambios durante una iteración

Por diferentes motivos, durante el curso de una iteración podría ser necesario incorporar alguna historia de usuario que sea de gran interés para la organización. En este caso no se pretende analizar los motivos en sí, sino brindar una alternativa de adaptación para que se pueda brindar una solución sin que ello comprometa de sobremanera el desarrollo de la iteración.

Considerando como punto de partida la necesidad de inclusión en la iteración actual de una historia de usuario, nueva o previamente generada, y la aceptación por parte del equipo de trabajo, se podrá proceder de la siguiente manera [94]:

1. Estimar la historia de usuario que se va a incorporar en los mismos términos (esfuerzo) que las actualmente integradas a la iteración en curso.
2. Determinar si su estado actual cumple con los requisitos de la DoR.
3. Seleccionar una historia de usuario que **no haya sido iniciada aún** cuya estimación sea igual o mayor a la que se pretende incluir.
4. Aclarar la situación con los interesados, cliente o su representante, indicando cuáles serían las consecuencias del cambio propuesto.

5. Proceder con el cambio realizando los ajustes necesarios en los diferentes gráficos, tableros y herramientas que se encuentre utilizando el equipo.

En caso de que ninguna historia de usuario no iniciada sea de igual o mayor valor de estimación que la que se pretende integrar se podrá realizar lo que se denomina *story splitting*. Esta técnica consiste en fraccionar la historia de usuario a incluir en otras de menor complejidad a fin de que la inclusión de la misma sea posible en la iteración en curso [71, 94]. En este caso, la priorización de qué fragmento incluir o no será tarea del representante del cliente dado que afectará los resultados potencialmente disponibles al final de la iteración.

Esta situación aplica a una iteración en la que se esté ejecutando cualquier fase del proceso de ciencia de datos, por ejemplo: integrar una nueva fuente de datos que no se había considerado hasta el momento en el análisis, generar una nueva característica para evaluar los datos desde otra perspectiva, integrar una dimensión de análisis que podría brindar una ventaja competitiva al negocio en el marco de una ventana de oportunidad, entre otros.

Problemas al cierre de una iteración

Puede suceder que al llegar el momento de cierre de una iteración, el desarrollo pactado para la misma no haya podido ser completado quedando historias de usuario o tareas técnicas pendientes de implementación. Nuevamente, no se pretende analizar los motivos entre las tareas que se proponen aquí, sino que eso deberá ser tratado en una reunión de retrospectiva.

La forma propuesta para proceder ante esta situación es:

- Comunicar la situación al cliente o su representante en forma previa, no esperar hasta la reunión de revisión de la iteración para hacerlo.
- Explicar los motivos por los que se ha llegado a tal situación, esto en la reunión de revisión de la iteración.
- Exponer que, aunque no es lo ideal, el haber ejecutado las historias de usuario en orden según su prioridad habrá resultado en que el producto implementado presente, potencialmente, las funcionalidades más importantes planificadas para su desarrollo en la iteración.

Entre las causas de estos retrasos se podrían mencionar: demoras en la adquisición de datos, problemas de calidad en los datos disponibles, modelos con efectividad más baja de lo esperado, entre otras.

4.5 Herramientas software de soporte a la propuesta

Con los aspectos procedimentales del marco de trabajo propuesto generados y descriptos en la sección previa, el próximo paso consistió en seleccionar una serie de herramientas software a emplear al momento de ejecutar y administrar un proyecto de ciencia de datos mediante la solución presentada. La lista que se detalla en este apartado es genérica y naturalmente deberá ser adaptada al entorno de trabajo de la organización que vaya a ejecutar un proyecto guiándose en el marco presentado.

La selección abarca a todo el conjunto de acciones incluidas en el proyecto, desde su fase de concepción hasta los procesos de despliegue del producto desarrollado para los usuarios finales. Adjunto a la selección realizada se encontrará una lista de ejemplos de productos software disponibles en el mercado para cumplir con la funcionalidad deseada al momento de la escritura del presente documento. No se incluyen herramientas ligadas estrictamente a las tareas de ciencia de datos debido al enfoque del trabajo con respecto al proceso presentado.

En general se requieren herramientas para las siguientes tareas:

- Gestión del *backlog* del producto y de las iteraciones, la trazabilidad de cada tarea y su asignación entre los integrantes del equipo.
- Gestión de las versiones del producto en desarrollo así como también los resultados intermedios que sean generados durante la ejecución del proyecto.
- Gestión de documentos compartidos entre todos los integrantes del equipo, por ejemplo: para los estándares de codificación, las definiciones aplicables a los conceptos de DoR y DoD, los valores de ciertos parámetros globales como la duración de cada iteración, entre otros.
- Gestión de entornos de trabajo para las diferentes instancias del proyecto, por ejemplo: desarrollo, pruebas, despliegue y producción.

- Gestión de automatizaciones para los flujos de integración y despliegue continuo.
- Gestión de servicios empleados para el despliegue en producción de la solución a desarrollar.

En función del listado previo, se describen en las siguientes secciones las herramientas a emplear.

4.5.1 Para la gestión del proyecto

En primer lugar, se requiere de una herramienta que permita realizar la gestión integral del proyecto a ejecutar. Para ello es necesario lo que usualmente se denomina como gestor de incidencias o tareas, para ser aplicable en este contexto deberá contar con las siguientes funcionalidades:

- Permitir la gestión del proyecto en base a iteraciones en forma nativa o vía complementos. Una plantilla que siga un enfoque similar al aplicable para el marco de trabajo Scrum sería recomendable.
- Permitir la definición de items de trabajo de diferente granularidad para lograr la jerarquía existente entre historias de usuario[6] y tareas técnicas.
- Permitir el registro y habilitar las funcionalidades a todo el equipo de trabajo con sus usuarios individuales para facilitar la asignación de tareas.
- Permitir la definición de ciertos atributos para cada item de trabajo como ser: un nombre, una descripción, fecha de inicio, la asignación de quién que deberá resolver o controlar la tarea en cuestión y su estimación en alguna unidad de medida. Adicionalmente sería deseable contar con la posibilidad de mencionar a los integrantes del equipo en su descripción o en una serie de comentarios, la disponibilidad para adjuntar archivos y el registro de las horas de trabajo dedicadas a la finalización del item.

[6]Existen casos en donde se aplica el concepto de épica como un requerimiento que no va a resolverse en una única iteración y se descompone en historias de usuario de menor tamaño.

- Permitir la definición de diferentes estados por los que podría pasar un item durante su ciclo de vida, siendo conveniente que tales etapas puedan ser definidas por el equipo del proyecto en función de como trabajen habitualmente. Mínimamente se debería poder contar con los siguientes estados: "pendiente", "en curso", "finalizado" (o similares).

Por otra parte, sería deseable contar con otras funcionalidades, sea de forma nativa, a través de complementos o mediante la vinculación con herramientas externas que permitan a los usuarios:

- Contar con un editor de documentos colaborativo del estilo *wiki* o similar para el registro de reportes, definiciones, acuerdos y demás documentación del proyecto que es conveniente tener en un espacio con acceso común a todos los integrantes del equipo tanto para su visualización como para su modificación.
- Integración con la herramienta de control de versiones, a ser descripta más adelante, a fin de que se pueda realizar la trazabilidad de qué cambios introducidos en el código del proyecto se generaron para dar solución a uno o más items de trabajo definidos.
- Visualización y exportación de gráficos que permitan observar diferentes aspectos de cada iteración o del proyecto mediante indicadores seleccionados por el equipo, por ejemplo: el gráfico de *burndown* de cada iteración.

Más allá de las características mencionadas se deben considerar dentro de los criterios de selección a las restricciones que pueda tener la organización en la que se va a ejecutar el proyecto. Principalmente, en lo que respecta al uso de una herramienta en forma *online* mediante un servicio en la nube pública (*SaaS*[7]) o si por el contrario, será necesario que la herramienta a utilizar se encuentre instalada en la infraestructura privada (sea en la nube o en forma local) de la organización.

[7] *Software as a Service* por su sigla en inglés.

Finalmente, ejemplos de este tipo de herramientas, al momento de la publicación de este documento, podrían ser: Jira[8], Redmine[9], Azure DevOps[10], Trello[11], entre otras.

4.5.2 Para la gestión del versionado del producto

En este caso, se trata de una herramienta aplicable para que los diferentes archivos del producto se puedan versionar de manera eficiente y con seguridad. Los elementos en cuestión no solo se limitan a código fuente sino que incluyen: documentación del proyecto, archivos de datos, configuraciones de entornos, entre otros [92]. La herramienta a utilizar deberá contar con las siguientes funcionalidades:

- Permitir generar un repositorio con el código del producto a versionar.
- Permitir registrar y almacenar cambios sobre los archivos involucrados (realizar un *commit*) y que tales cambios sean reversibles en caso de ser necesario.
- Permitir el registro de un conjunto de usuarios correspondientes a los integrantes del proyecto y brindar acceso al repositorio con diferentes restricciones.
- Permitir mantener varias líneas o ramas de desarrollo en paralelo y proveer las herramientas para su unificación.
- Permitir realizar revisiones de código e incluir comentarios que permitan abrir discusiones entre los integrantes del equipo.

Adicionalmente se listan características deseables para esta herramienta, sea que estén disponibles de forma nativa o mediante extensiones:

- Integración con la herramienta de gestión de tareas para identificar el conjunto de cambios que implementan una o más tareas dentro de una iteración permitiendo observar su trazabilidad.

[8]Jira: https://www.atlassian.com/es/software/jira
[9]Redmine: https://www.redmine.org/
[10]Azure DevOps: https://azure.microsoft.com/es-es/services/devops/
[11]Trello: https://trello.com/

- Integración con métodos de automatización para facilitar acciones como la ejecución de pruebas, integración y despliegue del producto.
- Provisión de controles que generen avisos sobre fallos comúnes de seguridad, por ejemplo: a nivel de dependencias del proyecto o a nivel de publicación de claves o *tokens* de acceso a servicios.

Más allá de las condiciones expuestas previamente, el equipo deberá determinar si utilizará una solución *on premise*, es decir, instalada en su propia infraestructura o una solución *on cloud*, disponible a través de una nube pública o privada (*SaaS*). En caso de seleccionar la segunda alternativa, se deberá considerar la limitante de permitir la definición de repositorios privados en caso de tratarse de una solución operativa en la nube pública. Esto se debe a que ciertos proyectos podrían tener acceso a datos sensibles o que deban ser protegidos por parte de la organización y no deberían ser accesibles, en forma directa o mediante la visualización del producto que los procese, desde la plataforma a emplear.

Para este tipo de herramientas, al momento de la publicación de este documento, se mencionan las siguientes alternativas: GitHub[12], GitLab[13], Azure Repos[14], BitBucket[15], entre otras.

4.5.3 Para la gestión de la automatización y los entornos de trabajo

En el ciclo de desarrollo del producto se podría requerir de la definición de entornos, entendiendo por tales a la configuración sobre la cual se va a estar desarrollando, integrando, probando o utilizando la solución. La determinación de los mismos y su definición particular serán tareas del equipo del proyecto a resolver en la iteración cero, sin embargo, es conveniente disponer de herramientas que gestionen tales ambientes efectivamente. En la actualidad, las tendencias de la industria apuntan al uso de IaC[16] como parte del enfoque de trabajo DevOps[17] [95, 97]: mediante archivos de configuración se

[12]GitHub: https://github.com/
[13]GitLab: https://about.gitlab.com/
[14]Azure Repos: https://azure.microsoft.com/es-es/services/devops/repos/
[15]BitBucket: https://bitbucket.org/
[16]*Infraestructura as Code* por su sigla en inglés.
[17]*Development and Operations* por su sigla en inglés.

establecen los parámetros para la generación de infraestructura y/o configuración para los entornos antes mencionados y se los incluye en el mismo control de versiones que al código del producto. Esto garantiza que todo el equipo trabaja sobre una misma base de configuración tanto en sus estaciones de trabajo como en las instancias de integración o posteriores. Además, esta estrategia favorece la automatización de acciones tales como CI/CD, con lo que se reduce el *time-to-market*. Este contexto es altamente compatible con la agilidad buscada en la propuesta dado que permite que el valor que se genera mediante las iteraciones del proyecto se encuentre disponible para su uso con menor demora.

Existen diferentes niveles de uso de estas herramientas por lo que su aplicación en un proyecto se recomienda que sea progresiva. En las iteraciones iniciales podrían tener un uso más reducido, que se incremente a medida que se cuenta con una primera versión del producto disponible para los usuarios finales. Por este motivo, se identifican dos tipos de recursos aplicables en este apartado:

- **Herramientas para gestión de entornos**: donde se pueden encontrar las diferentes alternativas para la definición de ambientes, la gestión de librerías o recursos a utilizar para el trabajo sobre el producto del proyecto. No implican directamente automatización y podrían ser un primer paso para que al menos el equipo completo desarrolle el producto sobre una misma configuración. Posteriormente se podrían incluir aspectos sobre la instancia de integración, pruebas o despliegue según las que decida definir el equipo.
- **Herramientas para la automatización**: aquellas que van a permitir la definición de flujos de trabajo mediante tareas a ejecutar ante ciertos eventos que actúen como "disparadores". En estos casos se podrían incluir cuestiones referidas al aprovisionamiento de infraestructura sobre algún entorno utilizando los elementos de configuración disponibles a partir de lo mencionado en el punto previo. En este caso, se tendrían a disposición diferentes indicadores, métricas de monitoreo y herramientas de depuración para identificar errores que pudieran presentarse en algunos de los pasos definidos.

Si bien el uso de estas herramientas no se puede considerar obligatorio, se recomiendan para todo tipo de proyectos en los que exista un producto software involucrado a fin de minimizar tiempos de despliegue, optimizar las operaciones involucradas

en las fases posteriores al desarrollo y disponer de una visión global de los resultados de tales acciones mediante alguna interfaz unificada.

Entre las funcionalidades deseables para estos tipos de herramientas se pueden mencionar:

- Generación de archivos de configuración que puedan ser versionados como cualquier otro componente del producto.
- Uso de formatos compatibles con las herramientas de escaneo de código de las plataformas de versionado para identificar potenciales problemas de seguridad.
- Reconocimiento del lenguaje y/o *frameworks* empleados por el equipo de trabajo para proponer alternativas de automatización adaptadas.
- Definición de eventos disparadores para dar inicio al ciclo de integración y/o despliegue.
- Disponibilidad para definir etapas por las que deberá pasar el producto y las acciones a ser ejecutadas en cada una de ellas, además de los flujos de notificación o comunicación para indicar que las operaciones hubieran finalizado exitosamente o con errores.
- Definición de métricas de evaluación de la calidad de la solución y de los flujos automatizados.
- Capacidad de acceso a registros o bitácoras (usualmente denominados *logs*) de las acciones ejecutadas para poder depurar errores en las diferentes etapas.
- Capacidad para identificar qué versión del producto se encuentra disponible en una instancia en particular en un momento determinado.

Ejemplos de este tipo de herramientas, al momento de la publicación de este documento, podrían ser: Jenkins[18], TravisCI[19], Jira CI/CD, Azure DevOps, GitHub Actions, GitLab CI/CD, entre otras.

[18]Jenkins: https://www.jenkins.io/
[19]TravisCI: https://travis-ci.org/

4.5.4 Resumen

En definitiva, la selección de herramientas para brindar soporte a la ejecución del proyecto no es trivial, existen diversas alternativas en el mercado a tener en cuenta. La mejor será, naturalmente, diferente de una organización a otra por sus características internas. Sin embargo, el foco en la selección debería estar en torno a uno de los objetivos relevados al inicio del presente capítulo: disponer en un mismo espacio de los requerimientos, su distribución en las diferentes iteraciones del proyecto, su descomposición en tareas de menor granularidad y la vinculación tanto con el código generado como los productos, intermedios o finales, que sean integrados y/o entregados a los usuarios finales en todas sus versiones.

Capítulo 5

Validación

En este capítulo se documenta la validación del proceso de gestión presentado en el apartado anterior. Se inicia por listar las opciones disponibles para realizar esta tarea en función de la literatura del área, definiendo así la estrategia a seguir. Luego se pasan a describir las instancias de validación realizadas, finalizando en una interpretación de los resultados obtenidos.

5.1 Introducción

La propuesta generada a través de este trabajo, se puede catalogar en un sentido genérico como un artefacto o producto de tecnología de la información dado que busca ser una herramienta para mejorar un aspecto del trabajo dentro de proyectos de ciencia de datos en un tipo específico de organizaciones. Este tipo de artefactos son definidos a nivel general como un conjunto de elementos. Son construcciones (mediante un vocabulario y símbolos específicos); modelos (generados mediante abstracciones y representaciones); métodos (materializados en algoritmos y prácticas); e instancias (prototipos o sistemas implementados) [98, 99].

En función de sus características particulares, se dispone de un conjunto de métodos para su evaluación [98]:

- **Observacional**:

  - Analizando el producto en el ambiente de negocio para el que fuera generado (caso de estudio).
  - Monitoreando los resultados de su utilización en múltiples proyectos (estudio de campo).

- **Analítico**:
  - Evaluando la estructura del artefacto en función de indicadores estáticos de calidad, como podría ser su complejidad (análisis estático).
  - Determinando si el producto se adapta a las características técnicas de arquitectura de sistemas de información (análisis de arquitectura).
  - Demostrando las capacidades de optimización del producto o encontrando indicios para tal operación (optimización).
  - Estudiando su comportamiento a través de indicadores dinámicos de calidad, como ser: rendimiento (análisis dinámico).

- **Experimental**:
  - Estudiando cualidades del producto, por ejemplo: usabilidad, en un entorno específico (experimento controlado).
  - Ejecutando el artefacto con datos generados artificialmente (simulación).

- **Pruebas**:
  - Ejecutando el producto a través de sus interfaces para descubrir errores e identificar defectos (pruebas funcionales de caja blanca[1]).
  - Llevando a cabo pruebas de cobertura de algún indicador en la implementación del artefacto (pruebas estructurales de caja blanca).

- **Descriptivo**:
  - Usando información desde una base de conocimiento para generar un argumento a favor de la utilización del producto generado (discusión).
  - Construyendo diferentes casos en los que se pueda probar la utilidad del artefacto (escenarios).

[1]Se denomina así, usualmente, a las pruebas de un producto software en las que se observa la ejecución de las instrucciones o procesos a medida que se van probando.

Considerando las limitaciones definidas por los autores y las características de la propuesta generada, la validación a conducir en la continuidad del documento será de los tipos observacional y analítica. Si bien el modelo de validación experimental se reconoce como aplicable, dadas las condiciones que son requeridas para su implementación, fue descartado para esta instancia de trabajo, quedando abierta la posibilidad de su aplicación en el futuro.

En la continuidad de este capítulo se inicia con el uso del marco de trabajo en un proyecto de ciencia de datos en un dominio de negocio específico implementando así un estudio de caso correspondiente al método observacional. Posteriormente, se analizan las características estáticas de la solución a través de un marco comparativo específico para metodologías de proyectos de explotación de información disponible en la literatura [16, 100]. Esta segunda etapa se corresponde al método analítico de validación.

5.2 Caso de estudio

Tal como se mencionó en el apartado anterior, la solución propuesta en el presente trabajo tuvo como primera instancia de validación un caso de estudio correspondiente con el método observacional de validación para productos de tecnología de la información [98].

El caso de validación consistió en la generación de un producto de datos para el análisis de la actividad de estudiantes en las cátedras del ciclo de nivelación (equivalente en otros planes de estudio a un curso de ingreso) a una carrera de grado universitaria. Para lograr esto, y dado el contexto de dictado virtual de las asignaturas, se toman datos del Aula Virtual (AV) de la facultad en cuestión. Puntualmente, se trabajó con los datos de actividad de cada usuario de tipo estudiante, en particular fecha y hora de inicio / cierre de sesión en la plataforma en los cursos generados para cada comisión de dictado de las cátedras involucradas.

El producto de datos a generar fue uno de visualización a modo de reporte de la información de los estudiantes en base a una clasificación establecida sobre la cantidad de días de inactividad en la plataforma virtual. Y la extensión del proyecto cubierta

para servir como caso de validación del presente trabajo abarcó hasta el desarrollo de la PoC del producto, sirviendo para verificar si constituye un punto de control para determinar la viabilidad del mismo.

Como preguntas a responder a través de la ejecución del caso de estudio se pueden mencionar:

- ¿La propuesta permite gestionar el flujo de trabajo de un proyecto de ciencia de datos desde su inicio hasta la generación de una PoC?
- ¿La propuesta se adapta correctamente a un entorno en particular para la gestión de un proyecto?
- ¿La propuesta de selección de herramientas permite un seguimiento de todos los aspectos de la ejecución del proyecto necesarios para el caso?

En las próximas secciones se describen los pasos de la aplicación de la propuesta de gestión para este proyecto de ciencia de datos. Y finalmente, se ensayan respuestas a las preguntas precedentes a fin de concluir esta instancia de validación.

5.2.1 Inicio del proyecto

La transición hacia un cursado virtual para las diferentes carreras de la Facultad de Ciencias Económicas (FCE) de la Universidad Nacional de Misiones (UNaM) durante el año 2020 implicó diversos cambios en la forma de seguimiento de las actividades de los estudiantes. Es de interés en este contexto, implementar métodos que permitan identificar a aquellos estudiantes que no mantienen un grado de participación constante en el cursado virtual de las asignaturas correspondientes al ciclo de nivelación para las carreras de grado de la FCE. Se consideran tales asignaturas porque conforman el punto de inicio de la vida universitaria de los ingresantes en el ciclo 2021 y podría ser analizado el cursado en su completitud de forma relativamente simple ya que el mismo se distribuye a lo largo de ocho semanas.

El objetivo es obtener una vista de los estudiantes y su grado de participación en el cursado para definir acciones que puedan orientarse específicamente a aquellos

que presenten problemas y así, tentativamente, mitigar el efecto que pueda producirse sobre su cursado.

Los datos necesarios para realizar este proyecto se pueden encontrar en los registros que realiza el AV sobre la participación de los usuarios registrados en la plataforma. De esta manera, se podría tener un informe sobre la participación de cada usuario con perfil estudiante sobre el curso de una o más asignaturas en un momento determinado.

Entre los indicadores a obtener sobre cada estudiante se pueden mencionar:

- Inactividad (medida en días desde su última conexión).
- Grado de participación en el cursado (definido a través de los diferentes recursos que sean empleados por los docentes[2]).

Como usuarios potenciales del producto se identifican:

- Las autoridades de la institución, en primera instancia la Secretaria Académica.
- Los docentes a cargo de una o más asignaturas.
- Los tutores que, sin ser docentes en tales cátedras, colaboran en las iniciativas de seguimiento de la trayectoria formativa de los estudiantes.

El equipo de trabajo se conforma por miembros de la Dirección de Tecnología para la Gestión (DTG) y de la Secretaría Académica de la FCE. La DTG se encarga de la gestión de los aspectos de tecnologías de la información para brindar soporte a las actividades de la facultad. El personal del área está compuesto por siete personas, distribuidas en las áreas: técnica, infraestructura, comunicaciones y sistemas. Para el desarrollo del proyecto en cuestión serán afectadas parcialmente tres personas:

- RB[3] para gestión de aspectos relacionados con el AV y administración de servicios.

[2]Cada equipo docente podría priorizar el uso de un tipo u otro de herramientas para fomentar la participación y/o realizar la evaluación de los estudiantes, por lo que este indicador deberá poder adaptarse a las características de cada curso.

[3]Se trata de las iniciales de las personas afectadas al proyecto.

- HH para gestión de la implementación de uno o más VPS[4], aspectos relacionados a comunicaciones y de automatización.
- MR para la implementación del producto de datos a desarrollar.

Como se ha mencionado, también se verá involucrado personal de la Secretaría Académica de la FCE en tanto que es la autoridad responsable por el dictado de las asignaturas que conforman el ciclo de nivelación para las carreras de grado de la facultad. En este caso se cuenta con la participación de dos personas que cumplirán el rol de interesados por parte del negocio (*stakeholders*):

- LN como parte del equipo docente de las asignaturas de ingreso y potencial usuario del producto. Se trata de una persona con un contacto más fluido con el equipo en el trabajo diario.
- SR como parte de los equipos de coordinación de las asignaturas de ingreso y con autoridad para cuestiones referidas a la evaluación del producto a desarrollar. En este caso, se trata de una persona con incidencia en las demos del producto para brindar su *feedback* en tales instancias.

5.2.1.1 Generación del backlog del producto

En primer lugar se listan, las historias de usuario[5] iniciales del proyecto con su respectiva estimación de complejidad[6]:

- **ID**: HU01
  - **Descripción**: Como coordinadora quiero ver la progresión de asistencias de los alumnos a todos los cursos del aula virtual para monitorear la tasa de actividad (alumnos activos).

[4] *Virtual Private Server* por su sigla en inglés. Son servidores virtualizados que operan en la infraestructura de una organización.

[5] En este caso el equipo ha optado por utilizar este método de registro de requerimientos por su facilidad de comunicación con el usuario final.

[6] En este caso se ha utilizado como escala una de tipo numérica, que se detalla al final de la expresión de la historia de usuario. La forma de establecer estos valores ha sido la descripta en la sección 4.4.2.1

  - **Complejidad**: 1

- **ID**: HU02
  - **Descripción**: Como docente quiero ver cuáles alumnos no están participando en los cursos a los que estoy vinculado para poder establecer contacto con ellos rápidamente y evaluar acciones a seguir.
  - **Complejidad**: 1

- **ID**: HU03
  - **Descripción**: Como coordinadora quiero ver y analizar los resultados de las diferentes actividades y el grado de cumplimiento de los alumnos para evaluar el progreso de cada curso de la plataforma.
  - **Complejidad**: 2

- **ID**: HU04
  - **Descripción**: Como docente quiero ver y analizar los resultados generales de participación de los alumnos en las diferentes actividades planteadas en los cursos a los que estoy vinculado.
  - **Complejidad**: 3

- **ID**: HU05
  - **Descripción**: Como tutor quiero obtener los datos de los alumnos de un conjunto de cursos que estén teniendo dificultades para poder coordinar acciones de apoyo.
  - **Complejidad**: 1

- **ID**: HU06
  - **Descripción**: Como docente / tutor / coordinador quiero ver los datos de los alumnos que están inactivos para poder comunicarme con ellos para establecer alguna estrategia de recuperación.
  - **Complejidad**: 2

- **ID**: HU07

  - **Descripción**: Como docente / tutor / coordinador quiero ver los datos censales de los alumnos inactivos para establecer algún tipo de estrategia de contención en los casos que aplique.
  - **Complejidad**: 5

La priorización, determinada por los interesados en el producto, se refleja en el orden de las historias de usuario listadas previamente.

En lo que respecta a las tareas de ciencia de datos vinculadas a este artefacto se establecen las siguientes definiciones:

- **Objetivos del proyecto**:
  - Determinar indicadores a calcular para el caracterizar la participación de los estudiantes.
  - Obtener los datos de participación de los usuarios de tipo estudiante en el AV durante el periodo de cursado de las asignaturas correspondientes al ciclo de nivelación de las carreras de grado de la FCE UNaM. Abarcar todas las comisiones de dictado.
  - Procesar tales datos para obtener indicadores de su participación en los cursos correspondientes.
  - Generar un *dashboard* que funcione como reporte general de los indicadores requeridos. Este será el **producto de datos** a generar inicialmente. Deberá ser accedido desde un navegador web y contar con mecanismos de autentificación para proteger el acceso a los datos en cuestión, especialmente los de tipo censales, que se catalogan como sensibles.
- **Orígenes de datos**:
  - El origen principal de los datos es la base de datos (BD) del AV de la facultad. En ella se encuentran los datos que permiten caracterizar la participación de cada usuario en los cursos y sus actividades.
  - El origen de los datos censales de cada estudiante será la BD del sistema de preinscripción que los mismos emplean para solicitar la inscripción a las diferentes carreras de la facultad.

- **Métricas de evaluación**:
  - M1: Cantidad de cursos del AV cuyos datos sean procesados sobre el total.
  - M2: Cantidad de perfiles de actividad de usuario generados sobre el total de cursantes.
  - M3: Cantidad de indicadores calculados y agregados al producto sobre el total de solicitados por los usuarios.
- **Umbrales de éxito**:
  - Para M1 el umbral de éxito del proyecto será del 100%.
  - Para M2 el umbral de éxito del proyecto será del 80%, esto se debe a que puede que existan estudiantes cuyo cursado no sea el primero de las asignaturas y los datos censales ya no se encuentren disponibles en la fuente definida.
  - Para M3 el umbral de éxito del proyecto será del 80% considerando los indicadores de mayor importancia.

5.2.1.2 Planificación de versiones

En relación a las tareas de gestión vinculadas a este artefacto, se define la duración de las iteraciones del proyecto y se estima la velocidad del equipo de trabajo. En primer término, se establece una duración semanal para las iteraciones de trabajo, principalmente por una cuestión de organización y de entrega frecuente de resultados en función de la duración del cursado. En cuanto a la velocidad, el equipo no cuenta con un valor establecido previamente, asi que se realiza una estimación inicial de la misma en nueve (9) *story points*[7] (SP) por iteración/semana. El cálculo realizado obedece a la siguiente relación: considerando una dedicación parcial de los miembros del equipo de tres días por semana laboral, se define un SP por persona/día. El valor de un SP diario fue obtenido luego de descomponer una historia de usuario en tareas técnicas y estimar el esfuerzo (en horas de trabajo) para implementarlas completamente obteniendo así una medida de trabajo diaria que se determinó equiparar con la unidad de estimación a emplear.

[7]Término en inglés para una unidad de estimación de trabajo por historias de usuario.

En lo que respecta a la planificación de alto nivel del desarrollo del producto de ciencia de datos, se decide abarcar solo la implementación de la PoC. En este sentido, se seleccionan las historias de usuario a involucrar en tal entregable tomando la priorización del backlog del producto generado, es así como la historia HU01 pasó a conformar las funcionalidades esperadas de la PoC. Esta decisión se toma entre todos los interesados en el proyecto para servir como hito de evaluación de la viabilidad del mismo.

Posteriormente, se procede con la estimación detallada a fin de poder establecer cuántas iteraciones son necesarias para su implementación. Como resultado, se asignan 14 SP que, por la velocidad del equipo definida con anterioridad, resultan en que la implementación de la PoC requiere de dos iteraciones. A estas se debe sumar la iteración cero para lograr la configuración general de las herramientas a emplear, por lo tanto, el total asciende a tres semanas de trabajo.

En resumen, la planificación de versiones inicial del proyecto queda conformada de la siguiente manera:

- **Iteración cero**. Duración: una semana.
- **Desarrollo de la PoC**:
  - Historias de usuario a desarrollar: HU01.
  - Cantidad de iteraciones: dos (2). Duración: dos semanas.
  - Total de SP: 14 (dejando margen para alguna actividad de *spike*[8] o *slack*[9] que pudieran requerirse).

[8]En entornos ágiles se denomina así a una tarea que tiene por objetivo servir para comprender la manera en la que deberá ser implementada otra. Suelen ser tareas de investigación sobre algún aspecto puntual de la implementación con el que el equipo no tenga experiencia. Se incluyen en una iteración y tienen un tiempo definido (*time-boxed*)

[9]Se denominan con este nombre a tareas que generalmente se incluyen en una iteración para pagar -resolver- deuda técnica, no guardan relación con el compromiso de una iteración de cara al cliente sino que tienen un resultado para el equipo.

5.2.1.3 Organización del trabajo

Este artefacto implica la realización de tareas de gestión orientadas al establecimiento de diversos valores y conceptos aplicables para el desarrollo del proyecto. El detalle de los mismos se encuentra a continuación:

- **Reuniones a emplear**: se utiliza a nivel interno la reunión diaria, aunque con frecuencia adaptada dado que la dedicación de los miembros del equipo de trabajo es parcial. Y a nivel general de la iteración las reuniones de: planificación, revisión y retrospectiva.
- ***Timebox* para las reuniones**: el tiempo definido para la reunión diaria es de 15 minutos, mientras que para el resto de las reuniones se establece en una hora.
- **DoR**: se considera que una historia de usuario se encuentra lista para pasar a ser implementada en una iteración cuando se especifican sus criterios de aceptación y se incluye alguna referencia respecto del tipo de resultado esperado por el cliente con respecto a la implementación de la misma.
- **DoD**: se considera finalizado el desarrollo de una historia de usuario cuando cuenta con los criterios de aceptación aprobados en su totalidad (o con las salvedades que pudiera decidir aceptar el cliente) y se integra al resto del producto desarrollado.
- **Interacción con el cliente**: durante una iteración se puede invitar a alguno de los *stakeholders* para reunirse con el equipo de trabajo en casos donde se requiera tomar alguna decisión en el plano funcional o no funcional con respecto al producto de datos. Estas reuniones serán coordinadas, en los casos donde sea posible, con anterioridad para evitar problemas de disponibilidad. El medio de comunicación inicial será el correo electrónico institucional o la telefonía interna de la unidad académica.

5.2.2 Iteración cero

Al tratarse de tareas técnicas se describen las acciones realizadas por el equipo de trabajo para la configuración general del proyecto:

- **Configuración de entornos**:
  - Se define un entorno de desarrollo (local a cada miembro del equipo), uno de integración (donde operará la solución de ETL[10]) y uno de producción (al que accederán los usuarios finales para la utilización del producto a desarrollar).
  - Se define el *stack* tecnológico[11] a utilizar inicialmente:
    * Desarrollo: lenguajes Python y SQL, librerías varias como ser: jupyterlab, pandas, sqlalchemy.
    * Integración: motor de BD MariaDB, Apache AirFlow como herramienta de automatización de flujos.
    * Producción: servidor web / herramienta a definir.
- **Configuración de la estructura del proyecto**:
  - Siguiendo los lineamientos generales dispuestos en la propuesta se implementa una estructura general del proyecto tal como se detalla en la figura 5.1. El directorio *code* contiene todos los archivos del código fuente del proyecto, segmentados según se trata de archivos de configuración, de libretas de experimentación, de código en lenguaje SQL, o *scripts* que pasan de una fase de experimentación a una de explotación (diferenciando si se trata de elementos de trabajo con datos en diferentes instancias o de la aplicación que será accesible para los usuarios). En el directorio *datasets* se encuentran resguardos de los datos que se procesan en las etapas de desarrollo / experimentación. Son tomados como una muestra que permite definir, probar y/o ajustar los flujos de tareas a implementar. Y finalmente, la carpeta *docs* contiene recursos relativos a los datos, su comprensión o estructura; al proyecto, ciertos documentos de referencia generales o de entendimiento de cuestiones del negocio que sea necesario tener presente para todo el equipo; y una ubicación de reportes con aquellos documentos que sean presentados al cliente durante el desarrollo más allá de la aplicación disponible.

[10] *Extract, Transform and Load* por su sigla en inglés. Se trata de las operaciones de recuperación, preparación y almacenamiento de los datos del proyecto en un espacio accesible por el equipo.

[11] Se denomina así al conjunto de herramientas software a emplear en un proyecto, el término *stack* hace referencia a que operan como una pila que se compone de varias capas relacionadas entre sí.

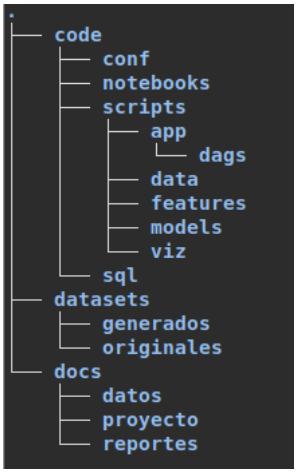

Figura 5.1: Estructura de directorios del proyecto.
Fuente: Elaboración propia

- **Especificar los estándares de codificación**:
  - En código que sea implementado en lenguaje Python se define como el estándar a utilizar a PEP8[12]. Para gestionar las versiones de las librerías a emplear se utilizan entornos virtuales y sus configuraciones se registran en el sistema de control de versiones.
  - En código que sea implementado en lenguaje SQL se deben utilizar los lineamientos expuestos en la guía de SQL de GitLab[13].
  - El repositorio de versionado se gestiona a través de ramas para las diferentes iteraciones o funcionalidades a incorporar según lo considere adecuado el equipo de trabajo.
- **Configuración del repositorio de versionado**:
  - Mediante un repositorio privado en la plataforma GitHub se define el repositorio de código del proyecto en cuestión. Para ello se ha decidido asignarle un nombre clave al proyecto, en este caso: "almendra".
  - Se generan los accesos al repositorio a los integrantes del proyecto con los permisos de escritura requeridos para poder realizar las modificaciones que se consideren necesarias.

[12]PEP8: https://www.python.org/dev/peps/pep-0008/.
[13]Guía SQL de GitLab: https://about.gitlab.com/handbook/business-technology/data-team/platform/sql-style-guide/

- **Definición de la arquitectura de la solución**:
  - El producto a generar se estructura, al menos para la PoC, de manera tal que cuente con un módulo encargado de realizar el proceso de ETL de los datos desde el AV de la facultad hasta una BD que podría denominarse de *stage*[14]. En otro módulo se encuentra la aplicación de visualización que toma los datos disponibles en la BD y presenta los gráficos del *dashboard* a los usuarios finales, esta aplicación debe tener implementado el control de acceso a usuarios registrados. Un diagrama de esta arquitectura se puede observar en la figura 5.2.

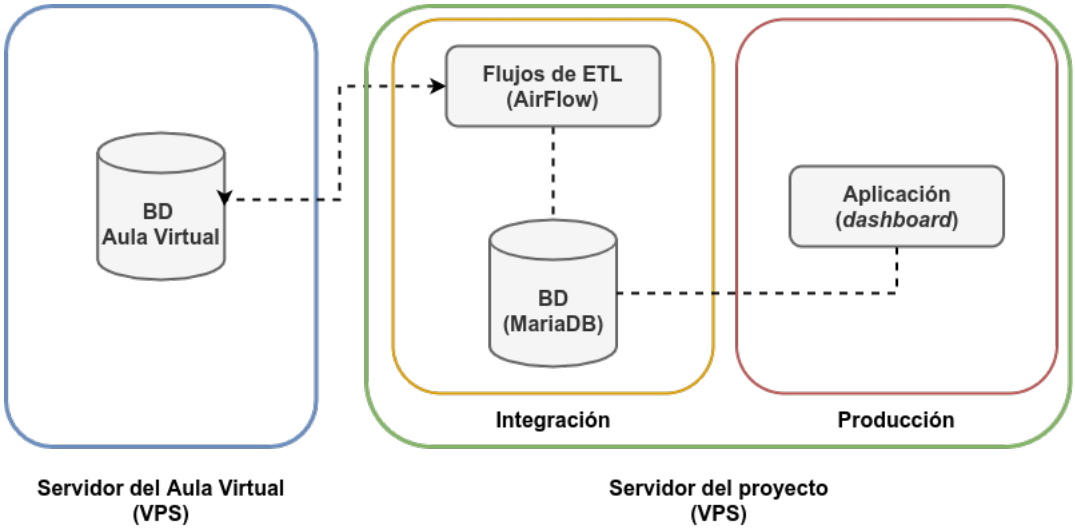

Figura 5.2: Arquitectura inicial del proyecto.
Fuente: Elaboración propia

- **Habilitación de la herramienta de seguimiento de incidencias**:
  - Para el seguimiento del proyecto se decide utilizar la herramienta Jira debido a que cumple con los requerimientos básicos solicitados en la propuesta. Se genera un proyecto basado en la plantilla de Scrum y se brinda acceso a todos los miembros del equipo de trabajo. Dada la estructura que provee la herramienta para la organización de los requisitos las historias de usuario se

[14]Se denomina de esta manera a la BD en la que se almacenan los datos crudos del proceso de ETL y sirve para la realización de pruebas. En este caso cumplirá la función de ser la BD de la PoC.

definen mediante items denominados *epics*[15]. El seguimiento de los mismos podrá realizarse a través de la herramienta de hoja de ruta (*roadmap*) y una historia podrá ser desarrollada durante más de una iteración. El siguiente nivel de detalle de los elementos de trabajo, los que se integran a cada iteración, son los items de *backlog* en la herramienta. Finalmente estos últimos se pueden descomponer en tareas de un mayor nivel de detalle. En todo momento se debe tratar de mantener la relación entre los elementos definidos para garantizar la trazabilidad en el desarrollo de la solución. En la figura 5.3 se puede ver una captura de pantalla de la definición de una historia de usuario como *epic*.

– Con respecto a las estimaciones, la herramienta provee un campo para determinar el valor de estimación en SP de cada item de trabajo definido. A fin de evitar confusiones se ingresan los valores correspondientes a las estimaciones de esfuerzo definidas para cada tarea a medida que los mismos son definidos.

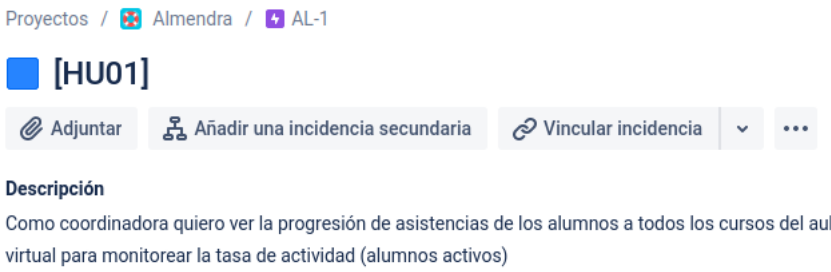

Figura 5.3: Historia de usuario del proyecto generada como épica en Jira.
Fuente: Elaboración propia

5.2.3 Iteración uno

5.2.3.1 Inicio de la iteración

Al tratarse de la primera iteración de trabajo del proyecto, no fue necesario realizar tareas de refinamiento del *backlog* del producto. Los siguientes pasos se describen en las líneas a continuación:

[15]Historias de usuario épicas, concepto que fue descripto previamente.

- **Selección de las historias de usuario para el *backlog* de la iteración**:
  - Descomposición de las historias de usuario en tareas técnicas: dado que en esta iteración se trabaja para el desarrollo de la PoC del proyecto el único requerimiento seleccionado es la HU01. Se procede con su descomposición en las tareas técnicas identificadas por el equipo de trabajo para lograr su implementación. Siguiendo con la jerarquía de items de trabajo que plantea la herramienta Jira, las tareas técnicas quedan definidas como items de *backlog* que se vinculan a la historia de usuario en cuestión (representada como épica en la herramienta como se ha mencionado). A fin de cumplir con la DoR en la *epic* se incluye un listado de criterios de aceptación y una descripción del resultado esperado por los *stakeholders* para su desarrollo.
  - Estimación detallada de las historias de usuario: utilizando como métrica de estimación los SP y como referencia lo mencionado al momento del cálculo de la velocidad del equipo, se realiza la estimación de esfuerzo de cada item de *backlog* definido para la HU01. El resultado de esta tarea y la anterior se pueden ver en la figura 5.4 que refleja los items de *backlog* definidos a partir de la historia épica con su respectiva estimación.

Con base en la ejecución de las dos tareas precedentes se observa que entre todas las tareas derivadas de la HU01 la cantidad de SP es superior a la velocidad del equipo. Por lo tanto, se debe seleccionar qué items de *backlog* incluir en la primera iteración y cuáles dejar para la segunda. El criterio de selección empleado fue la sucesión natural de las tareas necesarias para implementar la PoC y la cantidad de SP acumulados. De esta manera, se obtienen los items de *backlog* a desarrollar en la primera iteración:

- Generar proceso de ETL para los datos de actividad de los alumnos del aula virtual. Con una estimación de cuatro (4) SP.
- Automatizar la ejecución del proceso de ETL para garantizar la provisión de datos. Con una estimación de tres (3) SP.
- *Spike* para investigación de la herramienta AirFlow. Esta tarea se agrega dado que el equipo no tenía experiencia previa con la automatización de este tipo de tareas con la herramienta en cuestión. Se determina utilizar el margen de SP restantes de la iteración para poder conocer la herramienta y poder

implementar a través de ella la automatización del proceso de ETL. Este item se registra con una estimación de dos (2) SP.

Figura 5.4: Items de *backlog* vinculados a una épica en Jira.
Fuente: Elaboración propia

- **Armado del tablero de la iteración**:
  - Para el armado del tablero de la iteración se define el conjunto de tareas técnicas necesarias para resolver cada item del *backlog* de la iteración. Estas tareas de tercer nivel son las que se asignan entre los miembros del equipo de trabajo y, mediante su resolución, se implementa cada requerimiento. Se procede con la estimación de estas tareas con el mismo criterio aplicado previamente. Una muestra del resultado de esta actividad se puede visualizar en la figura 5.5 donde se detalla la descomposición de un item del listado de tareas de la iteración en sus tareas técnicas asociadas con su respectiva estimación.

Figura 5.5: Tareas vinculadas a un item de *backlog* en Jira.
Fuente: Elaboración propia

5.2.3.2 Desarrollo de la iteración

Esta primera iteración se enfoca en el desarrollo y automatización del proceso de ETL de los datos necesarios para el cálculo de los indicadores de actividad de los estudiantes. Se describen de las tareas involucradas en la ejecución de la iteración:

- **Generar proceso de ETL para los datos de actividad de los alumnos del aula virtual**:
  - Se analiza el modelo de datos de la BD del AV para identificar en qué tablas se almacenan los datos requeridos para el proyecto. Se realizan pruebas de extracción de datos para verificar las consultas SQL a implementar.
  - Se identifican los cursos de cuyos datos se debía realizar la extracción en esta instancia de trabajo. Esta operación se realiza desde la interfaz de usuario del

AV y permite obtener los valores necesarios para filtrar la extracción de datos solo a los cursos involucrados en la generación de la PoC del proyecto.

- Se implementan las consultas en lenguaje SQL para obtener los datos requeridos de la actividad de los estudiantes desde la BD del AV. Se prueban sobre un aplicativo de gestión de BD y se obtiene una vista de los datos con los que se estaría trabajando en el proceso de ETL. Se generaron dos consultas: una para los registros de usuarios de tipo estudiante inscriptos en un curso y otra para los registros de su actividad en la plataforma.
- Se toma un curso a modo de caso de prueba y se comienza a implementar el proceso completo de ETL con los siguientes pasos:
  1. Se realiza la extracción de los datos por medio de las consultas SQL generadas previamente.
  2. Se transforman los datos en cuestión a fin de unificar ambos orígenes de datos y obtener, sobre los registros de última conexión de cada usuario, la cantidad de días sin actividad a la fecha de extracción.
  3. Se almacenan temporalmente esos datos en un archivo csv[16] a modo de resguardo. Con este resultado intermedio se da continuidad a las siguientes tareas.
- Con los resultados de la ejecución parcial del proceso de ETL se procede con el diseño e implementación de la BD de *stage*.
- Se completa la actividad de carga del proceso de ETL haciendo que su ejecución escriba los datos resultantes en la BD de *stage*.
- Una vez testeado el proceso completo con el curso de prueba y verificado su correcto funcionamiento se generaliza su aplicación para todos los cursos involucrados en esta instancia.

- **Automatizar la ejecución del proceso de ETL para garantizar la provisión de datos**:
  - En primer lugar se pasa el código de la implementación del proceso de ETL de una libreta de experimentación a un *script* para facilitar su ejecución en forma automatizada.

[16] *Comma separated values*, valores separados por comas en inglés. Se trata de un tipo de archivo con valores organizados en filas y columnas que no tiene agregados de formato.

  - Se determinan los horarios y la frecuencia de ejecución del proceso. Diariamente y en un horario posterior al correspondiente a la última comisión de cursado (21hs) se programa la extracción de los datos.
  - Se implementa la automatización del proceso mediante la herramienta seleccionada, para ello se ejecutan previamente las tareas del *spike* que se definió en el último item de *backlog* de la iteración.
  - A modo de gestión de fallos y como resguardo, se determina que cada operación del proceso de ETL genere una salida a un archivo csv. Esto permite, ante una contingencia, restaurar la BD de stage con los datos intactos.

- ***Spike* para investigación de la herramienta AirFlow**:

  - Se destina tiempo para investigar de qué manera se puede automatizar la ejecución de código Python, en el que se implementó el proceso de ETL, mediante la herramienta AirFlow. Se analizan las opciones disponibles y se establece la forma en la que se efectiviza la automatización en cuestión.
  - En paralelo, se analiza la documentación de la herramienta para comprender de qué manera gestiona fechas y horas a fin de poder programar correctamente la ejecución de las tareas de ETL en los plazos y frecuencias previstas.

A lo largo del trabajo realizado se utilizan las diferentes herramientas consignadas previamente: los cambios sobre cualquier elemento del proyecto se sincronizan con el repositorio de GitHub del proyecto; las acciones que implicaron toma de decisiones sobre aspectos que pudieran afectar a los resultados a obtener se consultan con los *stakeholders* a fin de evitar conflictos; y los resultados de la ejecución de cada tarea se documentan, en los casos donde fuera oportuno, mediante comentarios agregados en las tareas en la herramienta Jira.

El estado final del tablero de la iteración se puede observar en la figura 5.6.

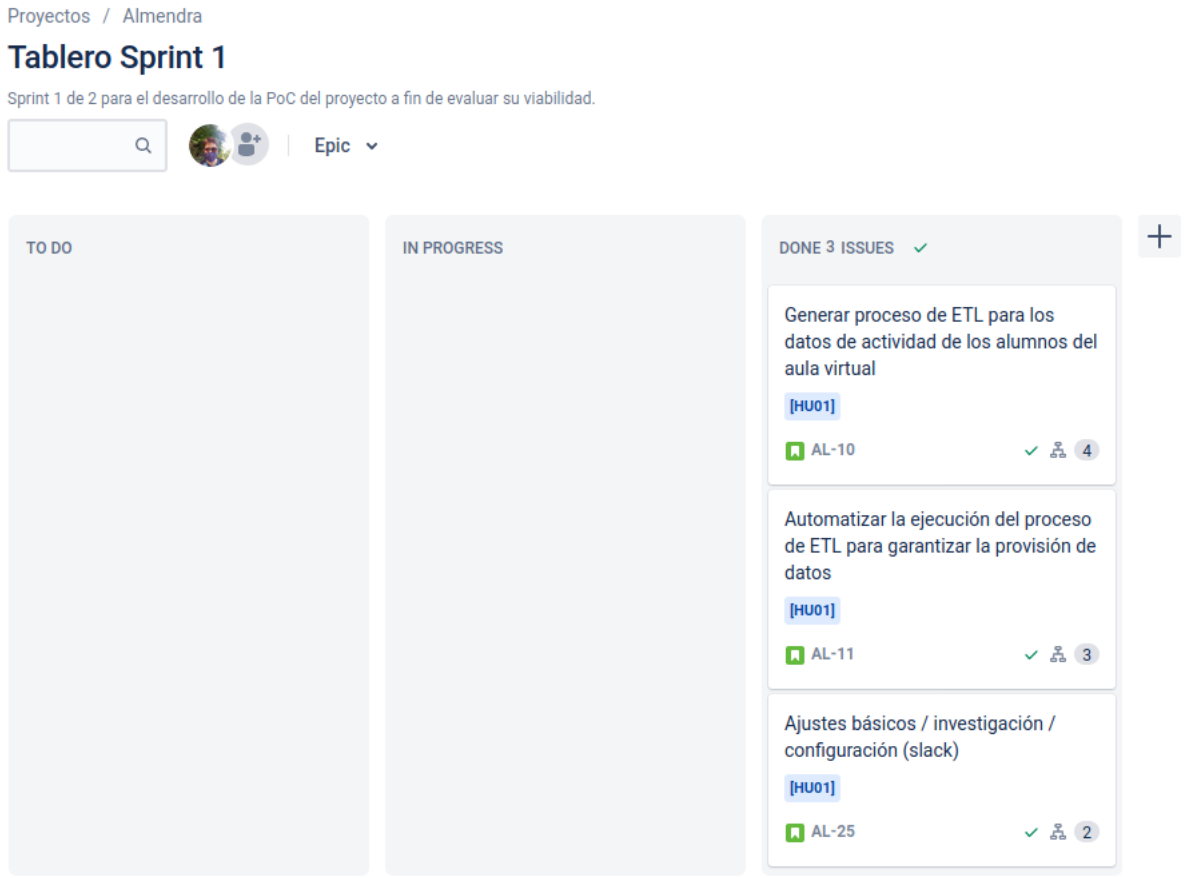

Figura 5.6: Vista del tablero al final de la primera iteración.
Fuente: Elaboración propia

5.2.3.3 Cierre de la iteración

Como resultado de la iteración se cuenta con el flujo de ETL implementado y automatizado para ser ejecutado con la frecuencia prevista. De esta manera se cumple el objetivo principal de la iteración por parte del equipo de trabajo. El resultado se reconoce poco aplicable para una presentación con los usuarios finales, por lo tanto, se genera una presentación en donde se explica el proceso y las ventajas de su implementación en la manera que fue realizada.

En la reunión de revisión de la iteración los *stakeholders* aprueban el resultado presentado sin mayores observaciones. Todos los items del *backlog* comprometidos por

el equipo se implementaron efectivamente. Mediante el *feedback* de los usuarios se identifica que, aunque el equipo tuvo en consideración los horarios de ejecución del proceso de ETL, no se tuvo en cuenta que el AV guarda sus datos en otro formato de zona horaria por lo que se detectan algunas diferencias entre las cantidades de estudiantes estimadas para el cursado y los presentados. El problema en cuestión se pasa a investigar en la próxima iteración a fin de realizar los ajustes necesarios en el procesamiento de los datos para evitar este tipo de situaciones. Como final de la reunión se planifican reuniones para la próxima semana en donde se propone aclarar cuestiones en torno a la clasificación de los estudiantes según su registro de actividad y los tipos de gráficos a generar. Se acuerda un día en particular y un rango horario para la misma para garantizar la participación de todos los involucrados en el proyecto.

En la reunión de retrospectiva, se analizan los procesos puestos en práctica en esta primera iteración por parte del equipo de trabajo. Se determina optimizar la comunicación con los *stakeholders*, en particular en aquellos casos donde se utiliza la telefonía interna de las instalaciones para aclarar dudas o cuestiones de interés para el equipo de trabajo. En tales situaciones se dispone reflejar lo hablado en un correo electrónico entre los involucrados y en la tarea afectada en la herramienta Jira mediante un comentario con las decisiones que se puedan tomar en base a la conversación sostenida. No se encuentran otras cuestiones a tratar por ser la primera reunión de este tipo y se termina por definir un item de *backlog* a trabajar en la siguiente iteración que involucra el análisis de los problemas detectados al finalizar la reunión de revisión.

De esta manera se concluye con la primera iteración del proyecto.

5.2.4 Iteración dos

5.2.4.1 Inicio de la iteración

La segunda iteración continúa el desarrollo de la historia de usuario planteada para la PoC por lo que, nuevamente, no se requiere realizar tareas de refinamiento del *backlog* del producto. Los siguientes pasos de esta fase se describen a continuación:

- **Selección de las historias de usuario para el *backlog* de la iteración**:

- Descomposición de las historias de usuario en tareas técnicas: se toman los resultados del trabajo realizado para la iteración anterior y se determina incluir al item generado como resultado de la reunión de revisión y retrospectiva de la iteración uno. El mismo se refiere a la investigación del modo en el que el AV almacena los datos de tipo fecha y hora, y cómo gestionar tales datos en el proceso de ETL.
- Estimación detallada de las historias de usuario: se refleja la estimación del el único item del *backlog* pendiente, el recientemente agregado, el resultado de ambas tareas se puede observar en la figura 5.7.

El item agregado para revisar las cuestiones detectadas en la iteración uno se estima con un valor que puede parecer alto, sin embargo esto se debe a que tiene una mayor carga de incertidumbre e implica tanto investigación como desarrollo posterior. Por este motivo la cantidad de SP estimados para la iteración se observa igual a la velocidad del equipo. Con el desarrollo de estos items, que definen la iteración dos, se estima finalizar el desarrollo de la HU01:

- Definir categorización o clasificación a emplear en los datos de alumnos para marcar actividad. Con una estimación de un SP y medio (1.5).
- Implementar una versión preliminar de los gráficos para los usuarios. Con una estimación de cinco (5) SP.
- *Slack* para investigación de la forma en la que el AV almacena las referencias de fecha y hora para así poder definir la estrategia a seguir para su procesamiento e implementarla. Este item se registra como resultado de las observaciones realizadas por los *stakeholders* en la reunión de revisión de la iteración anterior. La tarea se presenta con una estimación de dos SP y medio (2.5).

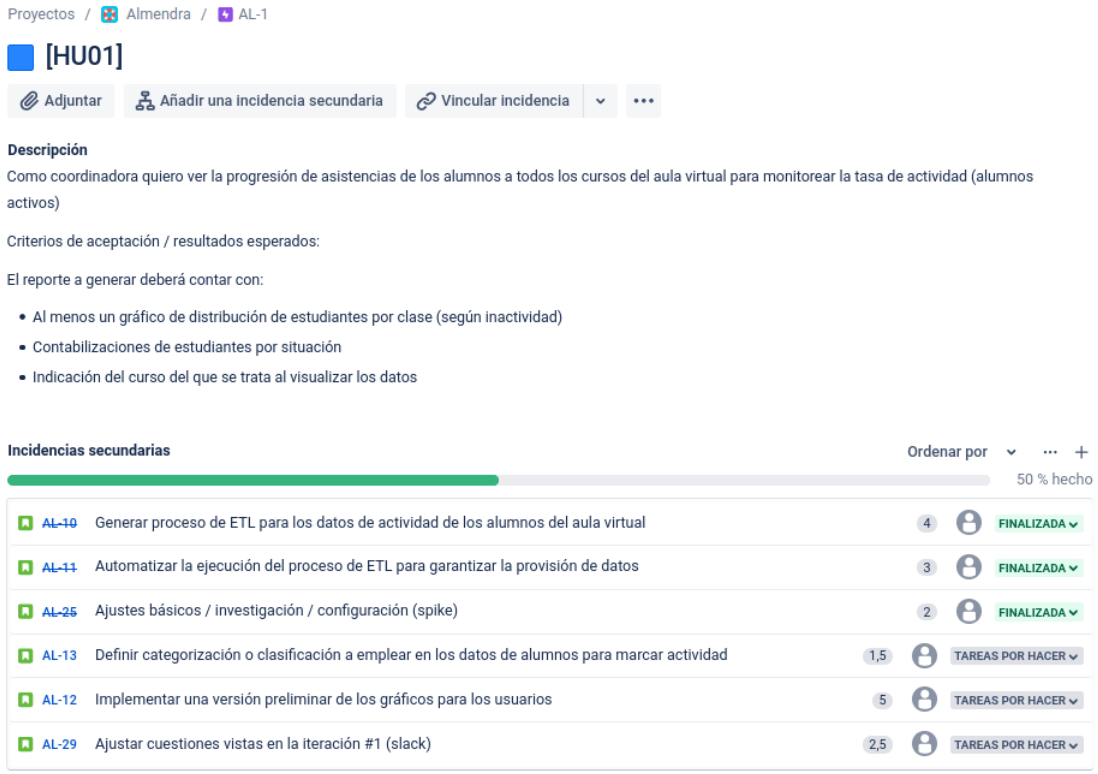

Figura 5.7: Items de *backlog* para la segunda iteración.
Fuente: Elaboración propia

- **Armado del tablero de la iteración**:
  - Se define el conjunto de tareas técnicas necesarias para resolver cada item del *backlog* de la iteración. Se procede con su estimación con el mismo criterio aplicado previamente. Una muestra del resultado de esta actividad se puede visualizar en la figura 5.8 que presenta la descomposición de un item en el listado de tareas técnicas asociadas para su resolución.

Figura 5.8: Tareas de un item de *backlog* para la segunda iteración.
Fuente: Elaboración propia

5.2.4.2 Desarrollo de la iteración

La segunda iteración se centra en completar el desarrollo de la PoC del proyecto. Contando con los datos en la instancia de *stage*, se busca determinar la clasificación a realizar sobre los datos de los estudiantes en función de su inactividad y generar el *dashboard* para presentar tales datos de forma gráfica a los usuarios finales. Se describen de las tareas involucradas en la ejecución de la iteración:

- **Definir categorización o clasificación a emplear en los datos de alumnos para marcar actividad**:
  - Se mantiene una reunión con los interesados en el proyecto para definir qué criterios se deben considerar en la clasificación de los estudiantes en función de los registros de inactividad. En la reunión se utiliza como insumo al conjunto de datos obtenido de la ejecución del proceso de ETL. Como resultado

de la misma se establecen los lineamientos para implementar la taxonomía en cuestión según los días de inactividad de cada usuario.

  - Se implementa el método de clasificación en una instancia de laboratorio y se prueba sobre datos disponibles en la BD de *stage* que se extraen a una fuente temporal. Las pruebas también se emplean para identificar en qué parte del proceso de ETL conviene aplicar la categorización en cuestión.
  - Pasada la fase de pruebas se agrega el código necesario para asignar la clase correspondiente a cada registro de estudiante en el paso de transformación del proceso de ETL. Además se realizan los ajustes necesarios en la estructura de la BD de *stage* para almacenar tales datos.
  - El cambio mencionado en el punto previo se aplica a las ejecuciones posteriores del proceso de ETL, sin embargo los datos previos en la BD se deben adaptar con igual criterio. Por tal motivo, se genera un conjunto de sentencias SQL que lo resuelven y, prueba mediante, se actualizan los valores de la BD para todos los registros de usuarios disponibles.

- **Implementar una versión preliminar de los gráficos para los usuarios**:

  - Se mantiene una reunión entre todos los involucrados en el proyecto para determinar los gráficos a ser integrados en la PoC. Se utilizan datos de la BD de *stage* para realizar planteos de diferentes alternativas por parte del equipo de trabajo y los *stakeholders* expresan sus opiniones al respecto.
  - Con los gráficos seleccionados, se pasa a implementar los mismos en un entorno de prueba a fin de identificar cuestiones a tener en cuenta y la forma adecuada de integrarlos en la PoC.
  - Se realiza un relevamiento breve para encontrar una herramienta que permitiera presentar un *dashboard* a partir de los gráficos a generar. Para esta instancia del proyecto se selecciona a Streamlit[17] como marco de presentación de los resultados en cuestión. La selección se basa en que se trata de una forma simple de implementar el tipo de reporte buscado y que se puede mejorar o migrar a otro contexto tecnológico con el avance del proyecto.
  - Con los resultados de las tareas anteriores se procede a implementar algunos de los gráficos seleccionados en la herramienta.

[17]Streamlit: streamlit.io

  - En simultáneo se trabaja sobre la verificación de las acciones que requiere el despliegue de la PoC con el contexto planteado.
  - Finalmente, se realiza el despliegue de la PoC sobre el entorno definido en la arquitectura inicial del proyecto y con la herramienta de visualización seleccionada.

- ***Slack* para ajustar cuestiones vistas en la iteración uno**:

  - En primer lugar, se realiza una revisión de la documentación del AV de la facultad para identificar el formato en el que se almacenan los datos de tipo fecha y hora (más que el formato, la zona horaria con la que son almacenados). A partir de esto se determinan los ajustes a implementar en la fase de transformación de los datos en el proceso de ETL para que los datos de la BD de *stage* reflejen la realidad. Esto se realiza a fin de evitar problemas al procesarlos tanto en las operaciones actuales como en las que se desarrollen a futuro.
  - En este item de trabajo se desarrolla, además, un método de resguardo complementario al planteado en la iteración anterior: se automatiza mediante otro flujo de AirFlow la realización de un *backup* de la BD de *stage*. El mismo se define para ejecutarse en forma diaria con anterioridad al proceso de ETL a fin de que, ante problemas de consistencia o integridad de los datos, la BD pueda ser restaurada.

Nuevamente, se marca el uso durante la iteración de las diferentes herramientas seleccionadas para dar soporte tanto al apartado de gestión como en el aspecto técnico del proyecto. El final de la iteración significa la implementación completa de la HU01, esto se ve reflejado en una captura de la hoja de ruta disponible mediante la herramienta Jira y representada en la figura 5.9. Una captura de pantalla de la PoC se puede observar en la figura 5.10 en donde se despliega uno de los gráficos con los datos de un curso en particular.

Figura 5.9: Hoja de ruta al final de la segunda iteración.
Fuente: Elaboración propia

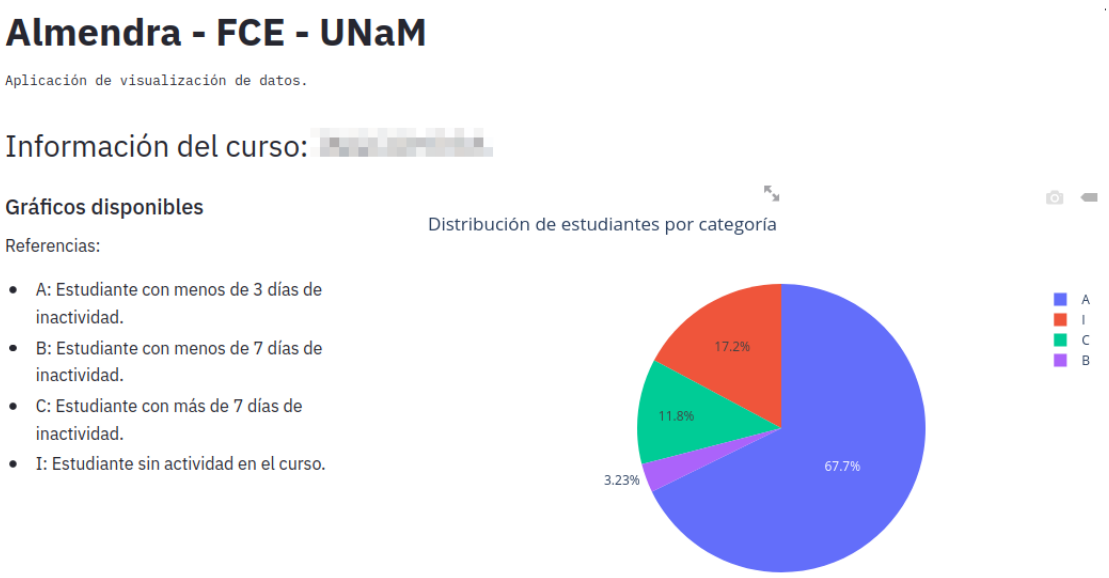

Figura 5.10: Vista de la PoC del proyecto en funcionamiento.
Fuente: Elaboración propia

5.2.4.3 Cierre de la iteración

Como resultado de esta segunda iteración se finaliza el desarrollo de la PoC del producto de datos definido para cumplir los objetivos del proyecto de ciencia de datos iniciado. Se realiza un despliegue de la misma que se presenta en la reunión de revisión.

En la reunión de revisión de la iteración los *stakeholders* observan el resultado del trabajo realizado, materializado a través de la PoC. La misma consiste en un reporte de la actividad/inactividad para los estudiantes de los cursos abarcados para esta etapa del proyecto. Aunque rudimentaria y sin algunas cuestiones no funcionales, que se postergan para las próximas iteraciones, se logra interactuar con la solución presentada. El *feedback* recibido por el equipo se considera positivo en general, con algunas observaciones en torno a la usabilidad, la disposición de los datos y la posibilidad de filtrar la vista disponible sobre un curso en particular. Todas cuestiones aceptables que el equipo aclara que se incluyen como parte de una versión posterior del producto, concretamente su MVP. Con el final de la reunión se resume la apreciación general sobre la PoC determinando su aprobación y, por lo tanto, marcando la viabilidad del proyecto al mismo tiempo que se confirma su continuidad. Se programa la próxima reunión de planificación de iteración para la siguiente semana y se finaliza la reunión.

En la reunión de retrospectiva, el equipo analiza los procesos utilizados en el proyecto después de dos iteraciones de trabajo. Los cambios propuestos en la reunión previa se analizan y se concluye que fueron adecuados para mejorar la comunicación entre todos los involucrados del proyecto. Como aspectos a considerar como resultado de la reunión de revisión se aceptan las cuestiones observadas y se determina que pasan a formar parte de las próximas iteraciones tal como estaba previsto originalmente. A modo de propuesta de mejora se incluye la determinación de que cuando se trate de la presentación de un aplicativo o similar, sea generado un prototipo básico para probar la interacción con los usuarios y minimizar los problemas de usabilidad y/o de presentación de información que pudieran producirse.

De esta manera finalizan las acciones de desarrollo del primer hito del proyecto y se aprueba su continuidad.

5.2.5 Análisis de resultados

Con la finalización del desarrollo del estudio de caso se procedió con el análisis de sus resultados a la luz de las preguntas que guiaron su ejecución. Se presentan las respuestas a las mismas:

¿La propuesta permite gestionar el flujo de trabajo de un proyecto de ciencia de datos desde su inicio hasta la generación de una PoC?

Se ha ejecutado una serie de iteraciones (una de configuración general y dos de trabajo) para un proyecto de ciencia de datos en un entorno real. Tanto las fases previas al inicio del trabajo como las del desarrollo del mismo en sí han sido gestionadas mediante el *framework* generado para el presente trabajo. Se han cubierto las actividades de gestión previstas y no se ha considerado que la carga de trabajo impuesta por la propuesta dificultara el avance del proyecto en cuestión.

La definición inicial del contexto de trabajo, las especificaciones y los acuerdos registrados por parte de todos los involucrados, han sido de ayuda para registrar la progresión del proyecto y sus resultados. Así como también las evidencias tanto del trabajo realizado como de las decisiones que se fueron tomando durante la ejecución de las iteraciones, todos constituyen elementos de seguimiento efectivamente implementados. En general, la inclusión de las herramientas de soporte para la gestión ha colaborado en la comunicación entre los integrantes del equipo de trabajo, más considerando la dedicación al proyecto planteada desde un inicio. En este sentido, la interacción se ha registrado por diversos medios y los avances no han sido afectados por las limitaciones mencionadas.

En relación a la iteración cero, se reconoce que su inclusión ha sido de utilidad en el contexto de ejecución del proyecto para evitar que una iteración de trabajo se vea afectada por la realización de ese tipo de tareas. De no contarse con esa instancia, el trabajo de preparación de entornos y demás podría consumir parte del tiempo de trabajo de una iteración. Y, al final de la misma, quizás no habría un resultado concreto a presentar como incremento del producto. Mientras que los elementos de gestión del cierre de una iteración han mostrado efectividad en lo que respecta a buscar una mejora continua de la calidad tanto del producto como de los procesos involucrados en el proyecto.

Los resultados obtenidos permitieron evaluar positivamente a la propuesta generada en este aspecto.

¿La propuesta se adapta correctamente a un entorno en particular para la gestión de un proyecto?

El caso de estudio ha representado justamente un caso en donde la solución planteada se debió adaptar a las particularidades del contexto. En primer lugar, el personal técnico afectado al proyecto no dispuso de una dedicación completa al mismo, tendiendo que alternar sus actividades habituales a las que requirió el desarrollo en cuestión. La selección de diversos recursos, como la forma de registro de los requisitos del proyecto, también han sido adaptados con éxito, contemplando desde la necesidad de los usuarios hasta las tareas técnicas para implementarlas en un producto. Los ajustes contextuales en torno al uso de recursos en las diferentes etapas del proyecto, como fueron las reuniones, su extensión y frecuencia; el modelo de planificación y hasta la selección de las herramientas para seguimiento del proyecto y sus derivados se puede considerar exitosa.

El desarrollo del caso de validación permite considerar que esta pregunta ha sido respondida de forma afirmativa.

¿La propuesta de selección de herramientas permite un seguimiento de todos los aspectos de la ejecución del proyecto necesarios para el caso?

En las respuestas previas se han mencionado aspectos de seguimiento que han sido resueltos de manera efectiva mediante los recursos planteados en el marco presentado. En este sentido la inclusión de las diferentes herramientas de soporte a la gestión mostraron ser de gran utilidad, especialmente en lo que refiere a la adaptación al contexto del proyecto en sí y al seguimiento de sus acciones y decisiones. También se estima que se logró un grado de documentación acorde. El mismo permitió garantizar la repetibilidad de las experimentaciones ejecutadas en el proyecto, más allá de que en este caso particular no se haya tratado de la generación de modelos de predicción mediante técnicas de aprendizaje automático por ejemplo.

El flujo de operaciones de la gestión del proyecto ha sido correctamente soportado por el conjunto de herramientas propuestas. Aún cuando no se avanzó en la integración de las mismas, cuestión que podría ser abordada en iteraciones posteriores.

Si bien en este caso no se han aplicado, por el alcance del desarrollo, las configuraciones o herramientas necesarias para implementar un flujo de CI/CD sobre la base de la selección realizada lo permitiría a futuro. Siendo una tarea a abordar por el equipo del proyecto una vez aprobada la PoC del mismo a fin de optimizar el uso de los recursos disponibles en las entregas de las futuras versiones del producto.

Mediante la experiencia del caso de estudio esta pregunta también se respondió en forma afirmativa.

De esta manera, se concluyó que la propuesta es adecuada para su aplicación en la gestión de proyectos de ciencia de datos mediante un conjunto de técnicas ágiles y el uso de una serie de herramientas software de soporte que se incorporan progresivamente al entorno de trabajo del equipo definido en función de los avances logrados.

5.3 Evaluación comparativa de la propuesta

En este apartado se evalúa la propuesta de gestión mediada por métodos ágiles para proyectos de ciencia de datos presentada en el capítulo cuatro haciendo uso de un marco comparativo de metodologías para este tipo de proyectos [100]. El *framework* generado se comparó con otros enfoques que se consideran los más relevantes de la industria.

El marco comparativo seleccionado para esta tarea se compone de cuatro elementos o aspectos [16, 99, 100]:

- Nivel de detalle en las actividades de cada fase, analizando la guía proporcionada al usuario en el desarrollo del proyecto.
- Escenarios de aplicación, evaluando la capacidad de adaptación a las necesidades del proyecto.
- Actividades específicas que componen cada fase, donde se observa el cubrimiento de las tareas para toda la extensión del proyecto.
- Actividades destinadas a la dirección de proyectos, analizando las actividades vinculadas a la gestión del proyecto disponibles.

En cada dimensión de análisis se considera una serie de propiedades que permiten valorar cuan abarcativa es la metodología o proceso analizado en tales aspectos. Se trata en total de 52 características a evaluar que tienen una respuesta binaria (si/no) y la alternativa con un mayor número de respuestas afirmativas se reconoce como mejor a las otras.

5.3.1 Comparación realizada

Para realizar la comparación fue necesario establecer contra qué metodologías sería evaluada la propuesta presentada. Es así que se tuvieron en cuenta las siguientes: el proceso de KDD, CRISP-DM y TDSP. La motivación detrás de esta selección fueron las respuestas obtenidas en dos encuestas online realizadas en los últimos años en donde se consultaba a diferentes actores de la industria qué metodología utilizaban para la gestión de sus proyectos de ciencia de datos [101, 102]. El producto aquí presentado, al que se le asignó la sigla MACD (Marco Ágil para Ciencia de Datos), sería el cuarto elemento en la comparación a realizar.

El paso siguiente consistió en calcular las características de cada aspecto planteado en el marco comparativo. Para las metodologías externas al presente trabajo se utilizaron los resultados disponibles en otros documentos donde se aplicara el mismo método de evaluación a fin de lograr mayor imparcialidad en la valoración [99]. Para determinar la calificación del MACD se respondieron progresivamente las preguntas relacionadas a cada propiedad de cada aspecto del marco de trabajo [100].

La evaluación de cada aspecto de la metodología de comparación, con sus respectivos subtotales y el total general, se reflejan en las siguientes tablas:

- Aspecto: Nivel de detalle en las actividades de cada fase. Tabla 5.1.
- Aspecto: Escenarios de aplicación. Tabla 5.2.
- Aspecto: Actividades específicas que componen cada fase. Se presenta una tabla por fase:
  - Análisis del problema. Tabla 5.3.

Característica	KDD	CRISP-DM	TDSP	MACD
¿Se definen actividades específicas para cada fase del proceso?	0	1	1	1
¿Se explicitan los pasos a seguir para llevar a cabo cada actividad?	0	1	1	0
¿Se definen las entradas de cada actividad?	0	0	0	1
¿Se definen las salidas de cada actividad?	1	1	1	1
¿Se provee una guía de buenas prácticas para cada una de las actividades específicas?	0	1	1	1
Subtotales	1	4	4	4

Tabla 5.1: Evaluación comparativa: Nivel de detalle en la descripción de las actividades. Fuente: elaboración propia con agregado de datos de [99].

  - Selección y preparación de los datos. Tabla 5.4.
  - Modelado. Tabla 5.5.
  - Evaluación. Tabla 5.6.
  - Implementación. Tabla 5.7.
  - Resumen del aspecto. Tabla 5.8.

- Aspecto: Actividades de dirección del proyecto. Se presenta una tabla por criterio:
  - Gestión del alcance. Tabla 5.9.
  - Gestión del tiempo. Tabla 5.10.
  - Gestión del costo. Tabla 5.11.
  - Gestión del equipo de trabajo. Tabla 5.12.
  - Gestión del riesgo. Tabla 5.13.
  - Resumen del aspecto. Tabla 5.14.
- Resultados generales (sumatoria de todos los aspectos). Tabla 5.15.

En todas las tablas el valor 1 representa la respuesta afirmativa a la pregunta, mientras que el valor 0 representa la respuesta negativa.

Característica	KDD	CRISP-DM	TDSP	MACD
¿Se especifican actividades para la definición y el análisis del problema u oportunidad con el cual colaborará la minería de datos?	1	1	1	1
¿Se consideran puntos de partida alternativos donde el usuario no refiere un problema sino que sólo desea explorar sus datos?	0	0	0	1
¿La metodología es independiente del dominio de aplicación?	1	1	1	1
¿La metodología es aplicable a proyectos de diferente tamaño?	1	1	1	0
Subtotales	3	3	3	3

Tabla 5.2: Evaluación comparativa: Escenarios de aplicación.
Fuente: elaboración propia con agregado de datos de [99].

Característica	KDD	CRISP-DM	TDSP	MACD
¿Se propone una evaluación general de la organización?	0	1	0	0
¿Se identifica al personal involucrado en el proyecto?	0	1	0	1
¿Se define el problema u oportunidad de negocio?	0	1	1	1
¿Se propone una evaluación de las fuentes de datos?	1	0	1	1
¿Se analizan todas las soluciones posibles al problema?	0	0	0	0
¿Se especifican los objetivos del proyecto?	0	1	1	1
¿Se define un criterio de éxito para el proyecto?	0	1	1	1
¿Se realiza una evaluación general de las técnicas de minería que podrían utilizarse?	1	1	1	0
¿Se especifica de qué forma el usuario utilizará el nuevo conocimiento?	0	0	0	1
Subtotales	2	6	5	6

Tabla 5.3: Evaluación comparativa: Actividades específicas que componen cada fase. Análisis del problema.
Fuente: elaboración propia con agregado de datos de [99].

Característica	KDD	CRISP-DM	TDSP	MACD
¿Se propone un análisis exploratorio inicial de los datos?	1	1	1	1
¿Se sugieren actividades para la limpieza de los datos?	1	1	1	1
¿Se contemplan actividades para la transformación de variables y la creación de atributos derivados?	1	1	1	1
¿Se realiza un análisis descriptivo final sobre los datos depurados?	0	0	0	0
¿Se verifica con el usuario la completitud del conjunto de datos final?	0	0	0	0
Subtotales	3	3	3	3

Tabla 5.4: Evaluación comparativa: Actividades específicas que componen cada fase. Selección y preparación de los datos.
Fuente: elaboración propia con agregado de datos de [99].

Característica	KDD	CRISP-DM	TDSP	MACD
¿Se efectúa una selección de las técnicas que se utilizarán?	1	1	1	1
¿Se planifica la forma en la que se evaluarán los resultados?	0	1	1	1
¿Se efectúa una evaluación inicial de los modelos obtenidos?	1	1	1	1
¿Se proveen directivas para el caso donde se dificulta el descubrimiento de patrones?	0	0	0	0
Subtotales	2	3	3	3

Tabla 5.5: Evaluación comparativa: Actividades específicas que componen cada fase. Modelado.
Fuente: elaboración propia con agregado de datos de [99].

Característica	KDD	CRISP-DM	TDSP	MACD
¿Se interpretan los modelos en función de los objetivos organizacionales?	1	1	1	1
¿Se comparan y ponderan los modelos obtenidos?	1	1	1	1
¿Se propone una revisión general del proceso?	0	1	0	1
¿Se proveen directivas para el caso donde ninguno de los modelos obtenidos resulta viable?	1	1	1	1
Subtotales	3	4	3	4

Tabla 5.6: Evaluación comparativa: Actividades específicas que componen cada fase. Evaluación.

Fuente: elaboración propia con agregado de datos de [99].

Característica	KDD	CRISP-DM	TDSP	MACD
¿Se planifica la implementación del nuevo conocimiento?	1	1	1	1
¿Se propone la creación de un programa de mantenimiento?	0	1	0	1
¿Se entrega al usuario un resumen del proyecto?	1	1	1	1
¿Se documenta la experiencia adquirida por el equipo de trabajo?	0	1	0	0
Subtotales	2	4	2	3

Tabla 5.7: Evaluación comparativa: Actividades específicas que componen cada fase. Implementación.

Fuente: elaboración propia con agregado de datos de [99].

Grupo de características	KDD	CRISP-DM	TDSP	MACD
Análisis del problema	2	6	5	6
Selección y preparación de los datos	3	3	3	3
Modelado	2	3	3	3
Evaluación	3	4	3	4
Implementación	2	4	2	3
Subtotales	12	20	16	19

Tabla 5.8: Evaluación comparativa: Actividades específicas que componen cada fase. Resumen.

Fuente: elaboración propia con agregado de datos de [99].

Característica	KDD	CRISP-DM	TDSP	MACD
¿Se propone la selección de los entregables que se generarán durante el proyecto?	0	1	0	1
¿Se especifican actividades de control del alcance?	0	0	0	1
Subtotales	0	1	0	2

Tabla 5.9: Evaluación comparativa: Actividades de dirección del proyecto. Gestión del alcance.

Fuente: elaboración propia con agregado de datos de [99].

Característica	KDD	CRISP-DM	TDSP	MACD
¿Se realiza una definición y secuenciación de las actividades que se ejecutarán durante el proyecto?	0	1	1	1
¿Se realiza una estimación de la duración de cada actividad?	0	1	0	1
¿Se construye un cronograma para el proyecto?	0	1	1	1
¿Existen actividades de control del cronograma?	0	0	0	1
Subtotales	0	3	2	4

Tabla 5.10: Evaluación comparativa: Actividades de dirección del proyecto. Gestión del tiempo

Fuente: elaboración propia con agregado de datos de [99].

Característica	KDD	CRISP-DM	TDSP	MACD
¿Se efectúa una estimación de los recursos afectados por cada actividad?	0	1	0	0
¿Se realiza una estimación de los costos del proyecto?	0	0	0	0
¿Se construye un presupuesto de costos?	0	0	0	0
¿Existen actividades de control del presupuesto a medida que avanza el proyecto?	0	0	0	0
Subtotales	0	1	0	0

Tabla 5.11: Evaluación comparativa: Actividades de dirección del proyecto. Gestión del costo

Fuente: elaboración propia con agregado de datos de [99].

Característica	KDD	CRISP-DM	TDSP	MACD
¿Se efectúa una planificación de los recursos humanos?	0	1	1	1
¿Se proponen actividades para motivar la interacción entre los miembros del Equipo?	0	0	0	1
¿Se efectúa un seguimiento del rendimiento de los recursos humanos?	0	0	0	0
Subtotales	0	1	1	2

Tabla 5.12: Evaluación comparativa: Actividades de dirección del proyecto. Gestión del equipo de trabajo

Fuente: elaboración propia con agregado de datos de [99].

Característica	KDD	CRISP-DM	TDSP	MACD
¿Se efectúa una identificación de los riesgos del proyecto?	0	1	0	0
¿Se realiza una cuantificación de los riesgos?	0	0	0	0
¿Se planifican acciones de respuesta ante cada riesgo?	0	1	0	0
¿Existen actividades de supervisión y control de los riesgos?	0	0	0	0
Subtotales	0	2	0	0

Tabla 5.13: Evaluación comparativa: Actividades de dirección del proyecto. Gestión del riesgo

Fuente: elaboración propia con agregado de datos de [99].

Grupo de características	KDD	CRISP-DM	TDSP	MACD
Gestión del alcance	0	1	0	2
Gestión del tiempo	0	3	2	4
Gestión del costo	0	1	0	0
Gestión del equipo de trabajo	0	1	1	2
Gestión del riesgo	0	2	0	0
Subtotales	0	8	3	8

Tabla 5.14: Evaluación comparativa: Actividades de dirección del proyecto. Resumen
Fuente: elaboración propia con agregado de datos de [99].

Aspectos (cantidad de características)	KDD	CRISP-DM	TDSP	MACD
Nivel de detalle en las actividades de cada fase (5)	1	4	4	4
Escenarios de aplicación (4)	3	3	3	3
Actividades específicas en cada fase (26)	12	20	16	19
Actividades para la dirección del proyecto (17)	0	8	3	8
Totales	16	35	26	34

Tabla 5.15: Evaluación comparativa: Resultados generales.
Fuente: elaboración propia con agregado de datos de [99].

5.3.2 Análisis de los resultados

La aplicación del marco comparativo de metodologías para proyectos de ciencia de datos [100] resultó en que el MACD propuesto en el presente trabajo alcanzó valores similares a la metodología CRISP-DM que se mantiene, a pesar del tiempo, como el estándar *de facto* en la industria. Si se analizan las dimensiones de análisis en forma independiente se pueden realizar las siguientes observaciones:

- **Respecto de la descripción de las actividades**: no se encuentran mayores diferencias con las metodologías CRISP-DM y TDSP, obteniendo el mismo puntaje total para el aspecto aún cuando el MACD no realiza una descripción detallada de cómo implementar cada actividad que propone realizar. Aunque sí cuenta con una definición de las entradas requeridas para las diferentes fases o actividades planteadas, siendo en varias ocasiones las salidas generadas de fases previas (por ejemplo: el backlog del producto se considera una entrada para la actividad de selección y estimación de las historias de usuario que pasarán a conformar el backlog de cada iteración).

- **Respecto de los escenarios de aplicación**: no existen mayores diferencias entre las cuatro alternativas analizadas. La particularidad de estar adaptado a pequeñas y medianas organizaciones hace que el MACD pierda generalidad en comparación a las demás opciones. Sin embargo, la posibilidad de que se analice el desarrollo de diferentes productos de datos para un problema en particular hace que permita definir puntos de partida alternativos para un proyecto que incluyen por ejemplo a la exploración de datos que menciona el marco comparativo.

- **Respecto de las actividades que componen cada fase**: en el apartado de selección y preparación de datos los resultados son equitativos. En el caso de MACD y la última propiedad sobre la verificación de la versión definitiva de los datos con el usuario corresponde especificar que, si bien no es una actividad explícita, podría aplicarse. Esto se sustenta en el hecho de que el equipo del proyecto integre a los usuarios finales del producto o sus representantes, y podría propiciar que se realice tal instancia de control. De igual manera si las tareas de selección y preparación de los datos fueran a ser presentadas en una reunión de revisión de iteración donde los *stakeholders* brindarían su opinión sobre las características del

dataset en cuestión. En las fases de modelado y evaluación no se observan diferencias significativas entre las alternativas evaluadas, mientras que en la etapa de implementación MACD no posee una actividad explícita de registro de las experiencias del proyecto en su totalidad, a pesar de eso la diferencia con la alternativa de mejor puntuación, CRISP-DM, es mínima (un punto).

- **Respecto de las actividades de dirección del proyecto**: en términos de gestión del alcance y del tiempo, la inclusión de métodos ágiles permite que MACD sume todos los puntos posibles en ambas categorías. En tales aspectos, las demás alternativas tienen resultados relativamente similares, excepto por el proceso KDD que no dispone de ninguna actividad de dirección del proyecto. En términos de gestión de costos y riesgos, la única opción analizada que tiene actividades relacionadas es CRISP-DM. En las acciones vinculadas al equipo de trabajo es donde tanto TDSP como MACD presentan actividades aplicables, con la reunión diaria como elemento que permite a esta última marcar una diferencia al constituir un espacio de interacción entre los miembros del equipo.

En definitiva, la propuesta del presente trabajo ha obtenido resultados similares a la opción de mayor puntaje y más utilizada a nivel general, CRISP-DM. Esto da pie a decir que MACD es una alternativa válida para la gestión de proyectos de ciencia de datos en el contexto para el cual fue definida. El *framework* cumple con la gestión de tiempos y alcances, aporta elementos de guía para la ejecución del proyecto y, a través de las instancias de trabajo conjunto con los interesados en el proyecto, permite analizar constantemente los avances logrados y adaptarse a cambios en el contexto de ejecución del proyecto en cuestión.

5.4 Resultados generales de la validación

Sobre la propuesta de proceso de gestión de proyectos de ciencia de datos con métodos ágiles se ejecutaron dos métodos de validación aplicables para el tipo de producto en cuestión [98]. Para el caso del método observacional, se desarrolló un caso de estudio a fin de llevar la aplicación del enfoque generado a un escenario real. En tal proyecto se ejecutaron tres iteraciones y se cumplieron las actividades necesarias para implementar

una PoC del producto de datos que se propuso desarrollar. Más allá del resultado de la evaluación de la PoC, el valor de la ejecución del proyecto bajo el proceso generado estuvo en la verificación de que en cada etapa se encontró una guía para la ejecución de las actividades planteadas. La adaptación necesaria al contexto de la organización también fue realizada de manera adecuada y permitió que seguir los lineamientos del marco no se convierta en un problema para el equipo del proyecto. Además de contar con un listado de herramientas para brindar soporte a la ejecución del proyecto permitiendo su seguimiento.

Para el caso del método de validación analítico, se utilizó un marco comparativo acorde a la solución presentada. Se siguieron los lineamientos para la evaluación contra metodologías que son ampliamente utilizadas en la industria utilizando valoraciones externas de las mismas para mayor independencia. Los resultados en este caso permitieron identificar al MACD como una alternativa válida para la gestión de proyectos de ciencia de datos. La integración de las técnicas ágiles de gestión para el proyecto y el flujo de tareas constituyó un diferencial frente a las otras opciones que, aunque más detalladas en algunos casos en términos de cómo ejecutar ciertas actividades, presentaron mayormente inconvenientes en los aspectos de gestión. Como resultado la validación analítica se consideró aprobada.

Con la salvedad de no haber ejecutado una estrategia de validación experimental por las limitaciones expuestas previamente, la propuesta de gestión se evaluó como adecuada para cumplir con los objetivos propuestos para el presente trabajo final de maestría.

Capítulo 6

Conclusiones

En este capítulo se publican las conclusiones a las que se ha llegado al final del desarrollo del presente proyecto, se analizan los resultados obtenidos a cada paso, a la luz de los objetivos planteados inicialmente y se mencionan los aportes generados. Por otra parte, se detallan problemas abiertos que podrían dar lugar a futuras lineas de investigación.

6.1 Conclusiones del trabajo

Al final del desarrollo del presente trabajo final de maestría se ha generado un *framework* para la gestión de proyectos de ciencia de datos mediado por técnicas ágiles. El mismo presenta diversos elementos adaptables para su aplicación en pequeñas y medianas organizaciones. Además provee un listado de herramientas software de soporte para su uso. Para arribar a este resultado se ha pasado por diferentes etapas que se describen a continuación:

- Se ha partido de sugerir una definición general de ciencia de datos para los límites del presente documento.
- Se relevaron y analizaron los tres grandes temas para construir el marco teórico del trabajo, en todos los casos considerando desde los enfoques más clásicos hasta los abordajes modernos:

  - Esquemas de gestión de proyectos de ciencia de datos.
  - Metodologías o modelos de procesos basados en agilidad para iniciativas de desarrollo de software.
  - Alternativas disponibles que apliquen técnicas o métodos ágiles a la resolución de problemas de ciencia de datos.

- Con esa base de conocimiento, se seleccionaron los criterios básicos que debería cumplir el marco a generar considerando las experiencias tanto de la academia como de la industria. A partir de ahí, se tomaron en cuenta dos cuestiones: en primera instancia las técnicas ágiles de mayor trascendencia y aplicabilidad para la propuesta a generar. Y en segundo lugar, los elementos en común entre las diferentes alternativas para la gestión de proyectos de ciencia de datos con técnicas o métodos ágiles, tanto a nivel de fases como de conceptos integrados.

- El paso siguiente fue definir el marco de trabajo que conformó la solución al problema enunciado. Se identificaron sus fases, la conexión entre ellas, las actividades a realizar y las técnicas aplicables en cada momento para llevar a cabo la ejecución de un proyecto de este tipo cumpliendo los criterios definidos y manteniendo su perfil ágil.

- Tal como se había previsto, la propuesta generada incluyó un conjunto de herramientas que, basadas en las tendencias y buenas prácticas de la industria, brindarían adecuado soporte al desarrollo de un producto de datos aplicando el enfoque presentado.

- Posteriormente, se abordó la validación de la solución utilizando para ello los métodos adecuados según la literatura del área. Se inició por la realización de un estudio de caso, desarrollando un proyecto en un entorno real y evaluando el desempeño del *framework* de gestión a cada paso. Y posteriormente, se realizó una evaluación analítica en donde se comparó la propuesta generada con las metodologías más utilizadas en el ambiente. En ambos casos la evaluación fue positiva.

De esta manera, se puede concluir que el producto resultante del presente trabajo final de maestría se considera aplicable para la gestión de proyectos de ciencia de datos mediante técnicas ágiles en pequeñas y medianas organizaciones. Es así como los objetivos propuestos inicialmente se pueden dar por cumplidos.

6.2 Futuras lineas de acción

Ya sea por las limitaciones del alcance del trabajo o por cuestiones detectadas durante su ejecución, se listan las posibles líneas de investigación a desarrollar con posterioridad:

- **Implementar métodos que permitan escalar la propuesta**: el enfoque presentado ha sido desarrollado para su adecuación al contexto de pequeñas y medianas organizaciones. Una posibilidad de trabajo a futuro consiste en la definición de la forma en la que el mismo podría ser aplicado a organizaciones y proyectos de una escala diferente. Esto implica cambios tanto en la gestión del equipo como en la gestión de las actividades del proyecto en sí mismo. Si bien existen alternativas para este tipo de situaciones en la actualidad, se trata de enfoques adaptados generalmente para el desarrollo de software y una reconversión para proyectos de ciencia de datos se considera necesaria.
- **Incluir un mayor nivel de detalle o guía en actividades de gestión clave**: puede suceder que un equipo con poca o nula experiencia en el desarrollo de proyectos de ciencia de datos no encuentre suficiente nivel de detalle en el *framework* propuesto. Más allá de que esto fue intencional, al no tratarse de una metodología, se podrían vincular diferentes estrategias disponibles en la literatura del área para resolver cuestiones puntuales. Algunas acciones en donde se observa esta posibilidad serían: la selección del tipo de producto a generar, la identificación del problema de ciencia de datos a resolver, entre otros.
- **Ampliar la selección de herramientas**: la selección de herramientas presentada para brindar soporte a la propuesta de gestión no abarcó a dos grupos: las específicas para tareas de ciencia de datos, y las relacionadas a la infraestructura de desarrollo o despliegue del producto. Esto se debió a que se buscó respetar la independencia de cada organización en la selección de su *stack* tecnológico sobre el cual trabajar. Sin embargo, se reconoce como un posible trabajo generar una estrategia de selección de herramientas específicas para actividades de ciencia de datos que considere diferentes entornos, productos y proveedores (principalmente a nivel *cloud*) según las características de cada proyecto.
- **Conducir una validación experimental**: quedando fuera de los límites del trabajo, una validación de este tipo se considera como un paso más en el sentido de la

verificación de su aplicabilidad. Para ello se deberían cumplir ciertas condiciones, entre ellas: dos equipos deberían ejecutar un mismo proyecto sin interacción entre sí; los equipos deberían contar con integrantes con diferentes niveles de formación y/o experiencia tanto en métodos ágiles como en ciencia de datos; un equipo debería gestionar el proyecto con un método a elección y otro mediante el marco propuesto; finalmente mediante un cuestionario común se evaluarían las opiniones de todos los involucrados a fin de verificar cuál enfoque resultó más útil o satisfactorio para ellos.

Bibliografía

[1] Cao, L.: Data Science: A Comprehensive Overview. ACM Comput. Surv. 50, 43:1–43:42 (2017).

[2] Donoho, D.: 50 Years of Data Science. Journal of Computational and Graphical Statistics. 26, 745–766 (2017).

[3] Tukey, J.W.: The Future of Data Analysis. Ann. Math. Statist. 33, 1–67 (1962).

[4] Naur, P.: Concise Survey of Computer Methods. Petrocelli Books, Studentlitteratur, Lund, Sweden (1974).

[5] Press, G.: A Very Short History Of Data Science, Sitio web: https://www.forbes.com/sites/gilpress/2013/05/28/a-very-short-history-of-data-science/. Último acceso: 08/10/2019.

[6] Fayyad, U., Piatetsky-Shapiro, G., Smyth, P.: From Data Mining to Knowledge Discovery in Databases. AI Magazine. 17, 37 (1996).

[7] Renae, S.: Data analytics: Crunching the future. Bloomberg Businessweek. September. Vol. 8, (2011).

[8] Berry, J.: Database Marketing, Sitio web: https://www.bloomberg.com/news/articles/1994-09-04/database-marketing. (1994). Último acceso: 08/10/2019.

[9] Davenport, T.H.: Competing on Analytics. Harvard Business Review. Enero. (2006).

[10] The Economist: Data, data everywhere, Sitio web: https://www.economist.com/special-report/2010/02/27/data-data-everywhere, (2010). Último acceso: 08/10/2019.

[11] McKinsey y Cia.: Hal Varian on how the Web challenges managers, Sitio web: https://www.mckinsey.com/industries/technology-media-and-telecommunications/our-insights/hal-varian-on-how-the-web-challenges-managers. (2009). Último acceso: 08/10/2019.

[12] Yau, N.: Rise of the Data Scientist, Sitio web: https://flowingdata.com/2009/06/04/rise-of-the-data-scientist/. (2009). Último acceso: 08/10/2019.

[13] Loukides, M.: What is data science?, Sitio web: https://www.oreilly.com/ideas/what-is-data-science. (2010). Último acceso: 08/10/2019.

[14] Britos, P.: Procesos de Explotación de Información Basados en Sistemas Inteligentes, Tesis de Doctorado. Doctorado en Ciencias Informáticas - Facultad de Informática - Universidad Nacional de La Plata (2008).

[15] Chapman, P., Clinton, J., Kerber, R., Khabaza, T., Reinartz, T., Shearer, C., Wirth, R.: CRISP-DM 1.0 Step-by-step data mining guide. CRISP-DM (2000).

[16] Moine, J.M., Gordillo, S.E., Haedo, A.S.: Análisis comparativo de metodologías para la gestión de proyectos de minería de datos. XVII Congreso Argentino de Ciencias de la Computación (CACIC 2011). ISBN 978-950-34-0756-1. (2011).

[17] Azevedo, A.I.R.L., Santos, M.F.: KDD, SEMMA and CRISP-DM: a parallel overview. IADS - DM. (2008).

[18] Pyle, D.: Business modeling and data mining. Morgan Kaufmann (2003).

[19] Wirth, R., Hipp, J.: CRISP-DM: Towards a standard process model for data mining. In: Proceedings of the 4th international conference on the practical applications of knowledge discovery and data mining. pp. 29–39. Citeseer (2000).

[20] Agile Alliance: Manifesto for Agile Software Development, http://agilemanifesto.org/iso/es/manifesto.html, (2000). Último acceso: 08/10/2019.

[21] Kwak, Y.H., Anbari, F.T. eds: The Story of Managing Projects: An Interdisciplinary Approach. Praeger, Westport, Conn (2005).

[22] Takeuchi, H., Nonaka, I.: The New New Product Development Game, https://hbr.org/1986/01/the-new-new-product-development-game, (1986). Último acceso: 08/10/2019.

[23] Schwaber, K.: Agile Project Management with Scrum. Microsoft Press, Redmond, Wash (2004).

[24] Sone, S.P.: Mapping agile project management practices to project management challenges for software development. Faculty of Argosy University/Washington DC College of Business, Doctor of Business Administration. (2008).

[25] Wysocki, R.K.: Effective Project Management: Traditional, Agile, Extreme. John Wiley & Sons (2011).

[26] Sutherland, J., Schwaber, K.: The Scrum Guide. scrumguides.org (2017). Último acceso: 08/10/2019.

[27] Sutherland, J., Sutherland, J.J.: Scrum: The Art of Doing Twice the Work in Half the Time. Currency, New York (2014).

[28] Kniberg, H., Skarin, M.: Kanban and Scrum - Making the Most of Both. Lulu.com, s. l. (2010).

[29] Beck, K.: Embracing change with extreme programming. Computer. 32, 70–77 (1999).

[30] Beck, K., Gamma, E.: Extreme Programming Explained: Embrace Change. Addison-Wesley Professional (2000).

[31] Poppendieck, M., Poppendieck, T.: Lean Software Development: An Agile Toolkit. Addison-Wesley (2003).

[32] Poppendieck, M., Cusumano, M.A.: Lean Software Development: A Tutorial. IEEE Software. 29, 26–32 (2012).

[33] Beck, K.: Test Driven Development. By Example. Addison Wesley, Boston (2002).

[34] Astels, D.: Test-Driven Development: A Practical Guide: A Practical Guide. Prentice Hall, Upper Saddle River, NJ (2003).

[35] Japan Management Association: Kanban Y Just In Time En Toyota.La Dirección Empieza En Las Estaciones. Prod. Press, Madrid, España; Portland, Or. (1998).

[36] Anderson, D.J.: Kanban. Blue Hole Press, Sequim, Washington (2010).

[37] Kniberg, H., Skarin, M.: Kanban and Scrum - Making the Most of Both. Lulu.com, s. l. (2010).

[38] Agile Alliance: What is Scrumban?, https://www.agilealliance.org/what-is-scrumban/. Último acceso: 23/10/2019.

[39] Ladas, C.: Scrumban: Essays on Kanban Systems for Lean Software Development. Modus Cooperandi Press (2011).

[40] Cockburn, A.: The heart of agile. Retrieved October. 25, 2017 (2014).

[41] Cockburn, A.: Heart of Agile, https://heartofagile.com/lets-begin/. Último acceso: 23/10/2019.

[42] Beck, K.: The Product Development Triathlon, https://www.facebook.com/notes/kent-beck/the-product-development-triathlon/1215075478525314/, Facebook. Último acceso: 23/10/2019.

[43] Beck, K.: Comparing Explore, Expand, and Extract: Topics in 3X, https://www.facebook.com/notes/kent-beck/comparing-explore-expand-and-extract-topics-in-3x/1241983035834558/, Facebook. Último acceso: 23/10/2019.

[44] Kerievsky, J.: An Introduction to Modern Agile, https://www.infoq.com/articles/modern-agile-intro/. Último acceso: 23/10/2019.

[45] Agile, M.: Modern Agile, http://www.modernagile.org/. Último acceso: 23/10/2019.

[46] Herger, B.: Project Management, for Data Science, https://medium.com/@13herger/project-management-for-data-science-3d9c53c0295b, (2018). Último acceso: 25/10/2019.

[47] Jurney, R.: Agile Data Science 2.0. O'Reilly Media, Inc. (2017).

[48] Jurney, R.: A manifesto for Agile data science, https://www.oreilly.com/ideas/a-manifesto-for-agile-data-science, (2017). Último acceso: 25/10/2019.

[49] Spacagna, G.: The Professional Data Science Manifesto, http://www.datasciencemanifesto.org/, (2015). Último acceso: 25/10/2019.

[50] IBM Analytics, "Analytics Solutions Unified Method. Implementations with Agile principles". IBM Corporation. (2015).

[51] Mérida Sánchez, J. C.: "Adaptación de Estándares de Dirección de Proyectos Particularizados para la Minería de Datos", Tesis de Maestría. Máster Interuniversitario en Gestión de Proyectos - Universidad de Oviedo (2017).

[52] Microsoft Corporation: ¿Qué es el Proceso de ciencia de datos en equipo (TDSP)? | Microsoft Docs, https://docs.microsoft.com/es-es/azure/machine-learning/team-data-science-process/overview, (2017). Último acceso: 25/10/2019.

[53] Microsoft Corporation: El ciclo de vida del proceso de ciencia de datos en equipo | Microsoft Docs, https://docs.microsoft.com/es-es/azure/machine-learning/team-data-science-process/lifecycle, (2017). Último acceso: 25/10/2019.

[54] Dolfing, H.: How to apply agile methods in data mining projects where CRISP-DM is used - Quora, https://www.quora.com/How-do-you-apply-agile-methods-in-data-mining-projects-where-CRISP-DM-is-used, (2016). Último acceso: 25/10/2019.

[55] Beckham, M.: Agile in Data Transformation, http://www.captechconsulting.com/blogs/agile-in-data-transformation, (2017). Último acceso: 25/10/2019.

[56] Raj, N.: CRISP-DM the Scrum Agile way. Why not!, https://www.virtulytix.com/intel/2018/3/15/crisp-dm-the-scrum-agile-way-why-not, (2018). Último acceso: 25/10/2019.

[57] Akred, J.: Agile Data Science Teams Deliver Real World Results, https://www.svds.com/agile-data-science-teams-deliver-real-world-results/, (2016). Último acceso: 25/10/2019.

[58] Akred, J.: Successful Data Teams are Agile and Cross-Functional, https://www.svds.com/tbt-successful-data-teams-are-agile-and-cross-functional/, (2016). Último acceso: 25/10/2019.

[59] Patil, D.J.: Building Data Science Teams - O'Reilly Media, https://www.oreilly.com/data/free/building-data-science-teams.csp, (2011). Último acceso: 25/10/2019.

[60] Digital.ai: State of Agile Survey 14th Edition (2020), https://stateofagile.com/. Último acceso: 16/07/2021.

[61] Data Science Process Alliance: Agile Data Science, https://www.datascience-pm.com/agile-data-science/. Último acceso: 16/07/2021.

[62] Lorica, B.: Using Agile development techniques for data science projects, https://www.oreilly.com/radar/podcast/using-agile-development-techniques-for-data-science-projects/. Último acceso: 16/07/2021.

[63] Nogueira, D.R.P.: Agile Data Mining: uma metodologia ágil para o desenvolvimento de projetos de data mining. (2014).

[64] Cristaldo, P.R., Schab, E.A., Richard, C.P., Rivera, R.A., De Battista, A.C., Retamar, M.S., Herrera, N.E.: Adecuación de una propuesta metodológica de enfoque "Híbrido" para la gestión de proyectos de ciencia de datos. (2018).

[65] Yan, E.: Data Science and Agile (What Works, and What Doesn't), https://eugeneyan.com/writing/data-science-and-agile-what-works-and-what-doesnt/. Último acceso: 16/07/2021.

[66] Yan, E.: Data Science and Agile (Frameworks for Effectiveness), https://eugeneyan.com/writing/data-science-and-agile-frameworks-for-effectiveness/. Último acceso: 16/07/2021.

[67] Atlassian: Jira Software, https://www.atlassian.com/es/software/jira. Último acceso: 16/07/2021.

[68] Azure Repos | Microsoft Azure, https://azure.microsoft.com/es-es/services/devops/repos/. Último acceso: 16/07/2021.

[69] Azure DevOps Services | Microsoft Azure, https://azure.microsoft.com/es-es/services/devops/. Último acceso: 16/07/2021.

[70] Agile Alliance: Subway Map to Agile Practices, https://www.agilealliance.org/agile101/subway-map-to-agile-practices/. Último acceso: 16/07/2021.

[71] Agile Alliance: Agile Glossary and Terminology, https://www.agilealliance.org/agile101/agile-glossary/. Último acceso: 16/07/2021.

[72] PMI, Project Management Institute: A Guide to the Project Management Body of Knowledge (PMBOK® Guide). Project Management Institute, Inc., USA (2017).

[73] do Nascimento, G.S., de Oliveira, A.A.: An Agile Knowledge Discovery in Databases Software Process. In: Xiang, Y., Pathan, M., Tao, X., and Wang, H. (eds.) Data and Knowledge Engineering. pp. 56–64. Springer, Berlin, Heidelberg (2012).

[74] Eclipse Foundation: Eclipse Process Framework Project (EPF) | The Eclipse Foundation, https://www.eclipse.org/epf/. Último acceso: 16/07/2021.

[75] Alnoukari, M., Alzoabi, Z., Hanna, S.: Applying adaptive software development (ASD) agile modeling on predictive data mining applications: ASD-DM methodology. In: 2008 International Symposium on Information Technology. pp. 1–6 (2008).

[76] Hadar, Y.: My Best Tips for Agile Data Science Research, https://www.kdnuggets.com/my-best-tips-for-agile-data-science-research.html/. Último acceso: 16/07/2021.

[77] Yan, E.: What I Love about Scrum for Data Science, https://eugeneyan.com/writing/what-i-love-about-scrum-for-data-science/. Último acceso: 16/07/2021.

[78] Nellutla, V.: Applying Agile IT Methodology to Data Science Projects, https://www.datasciencecentral.com/profiles/blogs/applying-agile-it-methodology-to-data-science-projects. Último acceso: 16/07/2021.

[79] Kashyap, N.: Data Science in Agile Mode: Rethinking the User Story, https://www.linkedin.com/pulse/data-science-agile-mode-rethinking-user-story-neelabh-kashyap. Último acceso: 16/07/2021.

[80] Thoen, E.: Agile Data Science. https://edwinth.github.io/ADSwR/. Último acceso: 16/07/2021.

[81] Martinez, D.: Do agile methodologies fit in data science environments?, https://datascience.aero/agile-methodologies-data-science/. Último acceso: 16/07/2021.

[82] AFIP: ¿Qué es una PyME? | Portal PyME, https://pymes.afip.gob.ar/estiloAFIP/pymes/ayuda/default.asp. Último acceso: 16/07/2021.

[83] CESSI: Reportes | CESSI Argentina, http://www.cessi.org.ar/opssi-reportes-949/index.html. Último acceso: 16/07/2021.

[84] Ministerio de Trabajo de la República Argentina: Observatorio de Empleo y Dinámica Empresarial (OEDE) | Estadísticas e indicadores nacional | Caracterización y evolución de la cantidad de empresas, https://www.trabajo.gob.ar/estadisticas/oede/index.asp. Último acceso: 16/07/2021.

[85] Biewald, L.:How to Build a Machine Learning Team When You Are Not Google or Facebook, https://www.kdnuggets.com/how-to-build-a-machine-learning-team-when-you-are-not-google-or-facebook.html/. Último acceso: 16/07/2021.

[86] Dhungana, S.:On Building Effective Data Science Teams, https://www.kdnuggets.com/on-building-effective-data-science-teams.html/. Último acceso: 16/07/2021.

[87] Altexsoft Inc.: How to Structure a Data Science Team: Key Models and Roles to Consider, https://www.altexsoft.com/blog/datascience/how-to-structure-data-science-team-key-models-and-roles/. Último acceso: 16/07/2021.

[88] Patil, D.J.: Building data science teams: the skills, tools and perspectives behind great data science groups. O'Reilly, Sebastopol, CA (2011).

[89] 1.Watts, S.: What is Sprint Zero? Sprint Zero Explained, https://www.bmc.com/blogs/sprint-zero/. Último acceso: 16/07/2021.

[90] Cookiecutter Data Science: Project structure | Cookiecutter Data Science, http://drivendata.github.io/cookiecutter-data-science/. Último acceso: 16/07/2021.

[91] Microsoft Corporation: TDSP Project Structure, and Documents and Artifact Templates, https://github.com/Azure/Azure-TDSP-ProjectTemplate. Último acceso: 16/07/2021.

[92] Software Engineering Course (SWEBOK) | IEEE Computer Society, https://www.computer.org/education/bodies-of-knowledge/software-engineering. Último acceso: 16/07/2021.

[93] Microsoft Corporation: Agile development of data science projects - Team Data Science Process - Azure Architecture Center, https://docs.microsoft.com/en-us/azure/architecture/data-science-process/agile-development. Último acceso: 16/07/2021.

[94] Paez, N., Fontdevila, D., Súarez, P., Fontela, C., Degiovannini, M., Molinari, A.: Construcción de software: una mirada ágil. , Buenos Aires (2014).

[95] RedHat: ¿Qué son la integración/distribución continuas (CI/CD)?, https://www.redhat.com/es/topics/devops/what-is-ci-cd. Último acceso: 16/07/2021.

[96] Amazon Web Services: Integración continua del software | Pruebas automatizadas | AWS, https://aws.amazon.com/es/devops/continuous-integration/. Último acceso: 16/07/2021.

[97] Atlassian: ¿En qué consiste la integración continua?, https://www.atlassian.com/es/continuous-delivery/continuous-integration. Último acceso: 16/07/2021.

[98] Hevner, A.R., March, S.T., Park, J., Ram, S.: Design Science in Information Systems Research. MIS Quarterly. 28, 75–105 (2004).

[99] Martins, S.: Modelo de proceso para proyectos de explotación de información. Tesis de Doctorado. Doctorado en Ciencias Informáticas - Facultad de Informática - Universidad Nacional de La Plata (2020).

[100] Moine, J.M.: Metodologías para el descubrimiento de conocimiento en bases de datos: un estudio comparativo, Tesis de Maestría. Maestría en Ingeniería de Software - Facultad de Informática - Universidad Nacional de La Plata (2013).

[101] Piatetsky, G.: CRISP-DM, still the top methodology for analytics, data mining, or data science projects, https://www.kdnuggets.com/crisp-dm-still-the-top-methodology-for-analytics-data-mining-or-data-science-projects.html/. Último acceso: 16/07/2021.

[102] Saltz, J.: CRISP-DM is Still the Most Popular Framework for Executing Data Science Projects, https://www.datascience-pm.com/crisp-dm-still-most-popular/. Último acceso: 16/07/2021.