



UNIVERSIDAD FASTA

Ingeniería en Informática – FIM 42 – Proyecto Final

Propuesta de Trabajo

# CommonJobs

(Sistema de Recursos Humanos CommonSense)

**Alumnos:**

- \* Andrés Moschini.
- \* Matías José.
- \* Juan Diego Raimondi.

**Director Funcional:** Gabriel Buyatti.

**Director Técnico:** Ing. Alejandro Fantini.

**Auditora:** Ing. Ana Haydee Di Iorio.

**Cátedra:**

- \* Profesor titular: AS. Hilario Fernando Schechtel
- \* Profesor asociado: Ing. Roberto Giordano Lerena
- \* Profesor asociado: Lic. Alejandro Nikolic

**Fecha de presentación:** 17/02/2012

## Tabla de Contenidos

Tabla de Contenidos .....	1
Introducción .....	2
Propósito de este documento .....	2
Alcance de este documento .....	2
Objetivo del proyecto .....	2
Descripción del Proyecto .....	2
Relevamiento .....	2
Proyecto .....	3
Alcance .....	3
Límites .....	3
Características Deseables .....	3
Características Mínimas (incluidas en el límite del proyecto) .....	3
Análisis FODA .....	3
Análisis Interno .....	3
Fortalezas .....	3
Debilidades .....	3
Análisis del Entorno .....	3
Oportunidades .....	3
Amenazas .....	4
Herramientas Disponibles .....	4
Proceso de Desarrollo .....	4
Product Backlog .....	4
Marca de Límite del Proyecto .....	4
Sprints .....	4
Entregables .....	5
Planificación .....	5
Gráfico de Gantt .....	5
Product Backlog (Preliminar) .....	6
Backlog Sprint 0 (Preliminar) .....	6

## Introducción

---

### Propósito de este documento

Este documento formaliza la Propuesta de Trabajo de los alumnos Andrés Moschini, Matías José y Juan Diego Raimondi, correspondiente al Proyecto Final de la Carrera Ingeniería en Informática de Universidad FASTA.

### Alcance de este documento

El presente documento está destinado a la Cátedra de Proyecto Final de la Facultad de Ingeniería de la Universidad FASTA, incluyendo al titular, asociados y auditores de la misma, al Director Técnico del proyecto Ing. Alejandro Fantini, al Director Funcional del proyecto Gabriel Buyatti, a los alumnos Andrés Moschini, Matías José y Juan Diego Raimondi, y al personal de *CommonSense Argentina*. Dicho documento establece las principales características del proyecto, junto con el marco en el que se desarrollará.

### Objetivo del proyecto

Brindar una herramienta para facilitar el trabajo del área de Recursos Humanos (RRHH) simplificando los procesos de carga, almacenamiento y lectura de la información manejada por dicha área e, indirectamente, optimizar sus resultados, mejorando la calidad de las incorporaciones a la empresa.

### Descripción del Proyecto

El proyecto consistirá en desarrollar e implantar un sistema informático que, principalmente, facilitará el seguimiento y la selección de candidatos a puestos de trabajo según sus capacidades, datos académicos, disponibilidad y pretensiones, junto con sus datos personales, información de contacto, e historial de contactos con la empresa. El sistema podría, además, facilitar otras tareas de RRHH, como generar las búsquedas de postulantes en sitios externos o mantener información de legajo de los empleados actuales.

### Relevamiento

*CommonSense Argentina* es una empresa de software nacida en la ciudad de Mar del Plata que brinda servicios de software y consultoría, también generando interés en la comunidad local de la educación en IT y promoviendo las buenas prácticas y aprendizaje académico. Surgió como una pequeña empresa con un producto propio y se ha extendido a ser una empresa de centenar de empleados en varios países (Argentina, Estados Unidos de América, México, Chile, El Salvador) aún promoviendo dichos servicios y asesoría de marketing digital.

La empresa está en este momento utilizando la herramienta [JIRA, de la compañía Atlassian](#), es una herramienta de seguimiento de proyectos que se ha extendido desde una herramienta de gestión de defectos. Esta herramienta está enfocada en el concepto de tareas que se crean y se comentan hasta que, tras pasar por determinados estados, llega a un estado final en donde se la da por terminada. Los estados son personalizables, pero el flujo de información es básicamente el mismo en todos los casos, y siempre se encuentra centrado en el concepto de una tarea. Teniendo esto en cuenta, la empresa mantiene los datos relacionados con las búsquedas laborales y postulantes de una manera informatizada pero no óptima, utilizando JIRA y hojas de cálculo Microsoft Excel de la siguiente manera:

- En el proyecto JIRA “Sales pipeline” otras áreas de la empresa inician tareas para las búsquedas de personal según sus necesidades.
- En el proyecto “Recruit”, por cada candidato se crea una tarea de JIRA y en campos personalizados almacena sus características, conocimientos, capacidades, CV, links a perfiles en redes sociales como LinkedIn y las entrevistas realizadas.
- En el proyecto “RRHH”, los empleados de la empresa inician tareas para el área de RRHH, básicamente pedidos de licencias o vacaciones.
- En un archivo de Microsoft Excel, se almacena el legajo de los empleados de la empresa, sus características, conocimientos, capacidades, sueldos, datos de contacto, etc.
- El área de RRHH responde a las solicitudes de “Sales pipeline” buscando los perfiles adecuados en el proyecto “Recruit” o generando búsquedas externas en publicaciones gráficas o sitios Web especializados como ZonaJobs, Boomeran o BuscoJobs.
- Los postulantes a las búsquedas externas realizadas son cargados en el proyecto “Recruit”.

Los principales problemas del sistema actual son los siguientes:

- JIRA no es la aplicación ideal para almacenar y buscar los datos de los candidatos: la carga o actualización es incómoda y artesanal. Si bien se utilizan campos personalizados, gran parte de la semántica se pierde, por lo cual las búsquedas ayudan muy poco en la selección.
- No hay integración con bases externas, muy utilizadas por la empresa, como LinkedIn, ZonaJobs, Boomeran, BuscoJobs, BaseJobs.

Dado el crecimiento actual y esperado de la empresa, a corto plazo este sistema no será sostenible: La información es volcada en JIRA, pero ese sistema no “entiende” el significado de la misma, la selección entre los candidatos almacenados depende de la memoria y *arte* de los empleados de RRHH.

La empresa analizó otras alternativas, que no se adaptaron a sus necesidades, entre ellas:

- [Zoho Recruit, Applicant Tracking System](#)
- [Staffing Soft, Recruiting Software](#)
- [CV Tracer, Advanced Recruiting Solutions](#)

Al ser *CommonSense* una empresa de software, se decidió construir un sistema informático propio para gestionar estos datos. Eso permitiría en un futuro expandirlo, para mantener el historial de capacitación y progreso de los propios empleados; facilitar la selección de los empleados óptimos para afrontar nuevos proyectos; u otras tareas relacionadas, o no, con el área de Recursos Humanos.

Aunque siempre estuvo presente en el espíritu de la empresa, el proyecto nunca se realizó. Por eso fue la elección obvia cuando se

consideró el tema con la dirección, para contar con su apoyo en el desarrollo de este proyecto final.

## Proyecto

---

### Alcance

- Desarrollo de un sistema de software para facilitar y optimizar las tareas del área de Recursos Humanos de *CommonSense*.
- Implantación del sistema a medida de que se vayan agregando características útiles al cliente.

### Límites

- No se contempla cubrir las necesidades de las otras áreas de la empresa más allá que las de RRHH.
- Dado que parte de las funcionalidades o características del sistema se decidirán durante el desarrollo del mismo, es importante dejar constancia en este documento que este proyecto comprende solo las tareas incluidas en el *Product Backlog* anteriores a la *Marca de Limite del Proyecto* (ver apartado Proceso de Desarrollo).

### Características Deseables

A priori identificamos las siguientes características deseables:

- Mantener información e histórico de los postulantes que hayan aplicado a la empresa.
- Agilizar las búsquedas en dicha información.
- Permitir buscar postulantes en bases de datos externas.
- Automatizar la extracción de parte de la información de los postulantes desde fuentes de datos externas como documentos de CV, sitios de búsqueda de empleo o redes sociales.
- Mantener el legajo de los empleados de la empresa con historial.
- Permitir generar búsquedas en publicaciones gráficas o sistemas de búsqueda de empleo.
- Integración con JIRA y otras herramientas utilizadas en el flujo de trabajo interno.
- Mantener información e histórico de los empleados actuales de la empresa.
- Ser acorde a los flujos que el área de RRHH desarrolla, los cuales pueden a su vez modificarse por la introducción de esta nueva herramienta.
- Capacidad de indicar automáticamente cuándo un postulante califica para determinada oferta de trabajo (como por ejemplo, si cumple con los requisitos mínimos).
- Permitir que otras áreas de la empresa puedan hacer pedidos de personal a RRHH.

De cualquier manera, podría ocurrir que no todas estén completas al momento de alcanzar la marca de “Limite del Proyecto” (ver apartado Proceso de Desarrollo). Por ejemplo, supongamos que alcanzado un grado de desarrollo importante el cliente propone, por caso, agregar mayor inteligencia aún a las búsquedas, tal vez eso insuma una cantidad de tiempo tal que el objetivo de generar búsquedas externas quede fuera del *limite del proyecto*. El propio dinamismo del proceso le permite al cliente re-priorizar las características que sean de mayor importancia para el desempeño su trabajo.

### Características Mínimas (incluidas en el *limite del proyecto*)

- Mantener información e histórico de los postulantes que hayan aplicado a la empresa.
- Registro de las entrevistas en la base de datos de postulantes
- Permitir búsquedas flexibles e inteligentes sobre la base de datos de postulantes.
- Mantener información e histórico de búsqueda de postulantes y ofertas de trabajo.
- Relacionar postulantes con búsquedas realizadas
- Seguridad para el acceso al sistema.
- Agilizar las búsquedas en dicha información.
- Migrar datos actuales (menos de 300 postulantes) del actual proyecto de Jira “Recruit” a nuestra base de datos

## Análisis FODA

---

### Análisis Interno

#### Fortalezas

- Los alumnos Andrés Moschini y Juan Diego Raimondi forman parte de la empresa *CommonSense*, conociendo los métodos internos y teniendo fácil acceso a información.
- Los alumnos ya poseen experiencia laboral en cuanto a desarrollo de aplicaciones y proyectos de software.
- La empresa posee licencias y herramientas que podrán ser utilizadas en el desarrollo del proyecto.
- Siendo la herramienta creada en base al proceso que *CommonSense* posee actualmente, su adopción será fácil.
- El proceso de implantación continua asegurará que el sistema satisface las necesidades del cliente.
- La característica del proyecto de ser en iteraciones permite la reconsideración de características a lo largo del proceso.

#### Debilidades

- El proceso de RRHH se ha ajustado según las herramientas disponibles, con lo cual puede que cambie al implementarse la nueva herramienta.
- Algunas de las tecnologías a utilizar serán nuevas para los miembros del equipo.

### Análisis del Entorno

## Oportunidades

- El desarrollo de este sistema podría permitir a *CommonSense* utilizarlo como base para futuras herramientas para mejorar otros procesos internos.
- Podría darse la oportunidad de, mediante algunas adaptaciones, empaquetar el producto para el uso de terceros.

## Amenazas

- La variabilidad en cuanto a las herramientas existentes en este área es tal que no existe una serie de buenas prácticas o procesos estándares en los cuales guiarse.
- Si el equipo no logra gestionar correctamente los requerimientos del cliente, el sistema podría estancarse y hasta fracasar. Lo mismo aplica para el caso del soporte.
- Podrían presentarse dificultades en el proyecto si alguno de los miembros o el director funcional abandona la empresa.

## Herramientas Disponibles

---

*CommonSense* desarrolla software utilizando principalmente tecnologías Microsoft y .NET. Por lo cual sería deseable utilizar básicamente herramientas relacionadas. A continuación detallaremos algunas de las que la empresa dispone y que podríamos utilizar, de adaptarse al proyecto:

- Servidores Internet Information Services (IIS)
- Servidores Microsoft SQL Server
- Atlassian JIRA
- Atlassian Confluence
- Licencias de Visual Studio 2010

## Proceso de Desarrollo

---

Dado que todos los integrantes del equipo tenemos una ocupación full-time y otras responsabilidades importantes extra-laborales, necesitamos una metodología flexible, que nos permita adaptarnos fácilmente a situaciones imprevistas externas e internas al proyecto y, a su vez, permitir fijar compromisos para periodos cortos de tiempo y evaluar el cumplimiento de dicho compromiso.

El cliente no necesita disponer del sistema de forma urgente, pero los sistemas que utiliza ahora mismo no son difíciles de reemplazar. Por otro lado, sabemos que dado el extenso tiempo del desarrollo de otros proyectos finales y de su implantación tardía, son poco los que son realmente utilizados finalmente por el cliente.

Dado el rápido crecimiento de la empresa y la creación relativamente reciente del área de recursos humanos, sus necesidades aún no están muy claras y es posible que vayan cambiando durante el desarrollo del sistema.

En varios proyectos de la empresa se está implementando el marco de trabajo *Scrum*, el cual no es aplicable a este proyecto por las siguientes razones:

- Al ser un grupo chico y los tres integrantes estar comprometidos - *Pigs* en la nomenclatura de *Scrum* - con el proyecto pero sin poder darle dedicación exclusiva sería un limitante definir un *ScrumMaster*.
- Si bien el *Product Owner* está interesado en el desarrollo del sistema, bien puede continuar su labor sin él, por lo tanto solo estaría involucrado - *Chicken*, según los usos de *Scrum* ya que no está comprometido.
- Tanto el esparcimiento del equipo (en distancias y zona horaria) como la dedicación parcial y en ratos libres de la ocupación laboral principal de los integrantes impedirán la realización de una reunión diaria.

Por todo lo anterior decidimos utilizar una metodología propia iterativa e incremental. Por familiaridad y comodidad utilizaremos la nomenclatura y algunos de los procedimientos de *Scrum*.

## Product Backlog

Contaremos con un *Product Backlog* donde identificaremos todos los requerimientos y funcionalidades deseables, priorizadas y con estimaciones de esfuerzo aproximadas (que iremos ajustando luego de realizados una serie de *sprints*). Ésta lista estará abierta, tanto el cliente, la cátedra o el equipo podrá incluir ítems, aunque la prioridad deberá ser negociada teniendo en cuenta la relación costo / beneficio según objetivo del sistema, los tiempos y las cuestiones técnicas.

## Marca de Límite del Proyecto

Se trazará una línea dividiendo las tareas del *Product Backlog* que representará cuales son las tareas que se espera realizar antes de la finalización del proyecto. La posición de esa línea se irá ajustando entre los *sprints* y al incluir nuevas tareas al *Product Backlog*.

De esta manera, tanto el cliente, el equipo y la cátedra tendrán una idea aproximada (y cada vez mas cercana a la realidad) de cómo será el resultado final. Y facilitará la fijación de prioridades a las tareas.

La posición de la *Marca de Límite del Proyecto* contemplará:

- Que se cumplan los objetivos mínimos del sistema (ver sección sobre Objetivos Específicos incluidos en el límite del proyecto).
- Que la fecha estimada para alcanzarla se encuentre en un rango de  $\pm 3$  meses del día 11 de marzo de 2013.
- Que la suma de las horas trabajadas por los integrantes del equipo (en su totalidad) esté en el rango de  $1500 \pm 200$  horas.

La única condición flexible en este caso es la que define los objetivos mínimos del sistema. Esto es a drede, ya que le permite al cliente realmente priorizar y determinar cuál es el punto en el cual él mismo determina en qué punto el sistema puede ser aceptado como final.

## Sprints

El *Sprint* es el periodo en el cual se lleva a cabo el trabajo en sí. Tendrán una duración fija, en principio de tres semanas pero podrá ser modificada durante el desarrollo del proyecto. Luego de comenzado un *sprint* no se podrán modificar sus tareas, con la sola excepción de tareas de soporte.

En la *Reunión de Planificación del Sprint*, cada integrante se comprometerá a trabajar una cierta cantidad de horas y se definirá el *Sprint Backlog*. En este documento se detallará cuáles son las características que se implementarán como los requisitos del *sprint* y se asignarán horas a las tareas correspondientes, intentando crear tareas de no más de 10 horas.

Al terminar se realizará la *Reunión de Revisión del Sprint*. Allí se revisará el trabajo completado y el no completado; se realizará una demo del trabajo completado a los interesados. Luego, los integrantes del equipo realizarán la *Retrospectiva del Sprint* intentando realizar una mejora del proceso y relevando las horas trabajadas realmente por cada integrante del equipo contrastándolas con las horas pactadas al inicio del *sprint*.

Luego de realizados una serie de *sprints*, a medida de que vayamos conociendo el ritmo del equipo y se vaya definiendo con mayor exactitud las tareas del *Product Backlog*, iremos realizando re-estimaciones del mismo y ajustando la posición de la *Marca de Límite del Proyecto*.

## Entregables

Serán entregados a la cátedra los siguientes documentos:

- Copia del *Product Backlog* al inicio de cada *sprint*
  - User stories (mas detallados a medida que avance el proceso)
  - Tareas técnicas
  - Estimaciones aproximadas
  - *Marca de límite de proyecto*
- Resultados de la *reunión de planificación de los sprints (Sprints Backlogs)*
  - Planificación del *sprint*
  - Horas comprometidas a trabajar por cada miembro
- Resultados de las *reuniones de revisión de los sprints*
  - Detalle de las tareas realmente realizadas
- Resultados de las *reuniones de retrospectiva de los sprints*
  - Mejoras y oportunidades
  - Horas realmente trabajadas por cada miembro discriminadas según el tipo de tarea (relevamiento, test, etc).
- Output de los *sprints* - Dada la variada naturaleza que cada *sprint* puede tener, cada uno de ellos podrá tener distintas salidas según el trabajo que requiera ser realizado. Identificamos estos como los más comunes a lo largo de la ejecución del proyecto:
  - Documentos comparativos acerca de posibles tecnologías o soluciones
  - Documentos de análisis y diseño pertinentes, con el nivel de detalle requerido en cada caso
  - Código fuente y ejecutables de pruebas de concepto
  - Código fuente y ejecutables del sistema
  - Registro de la realización de tarea de soporte

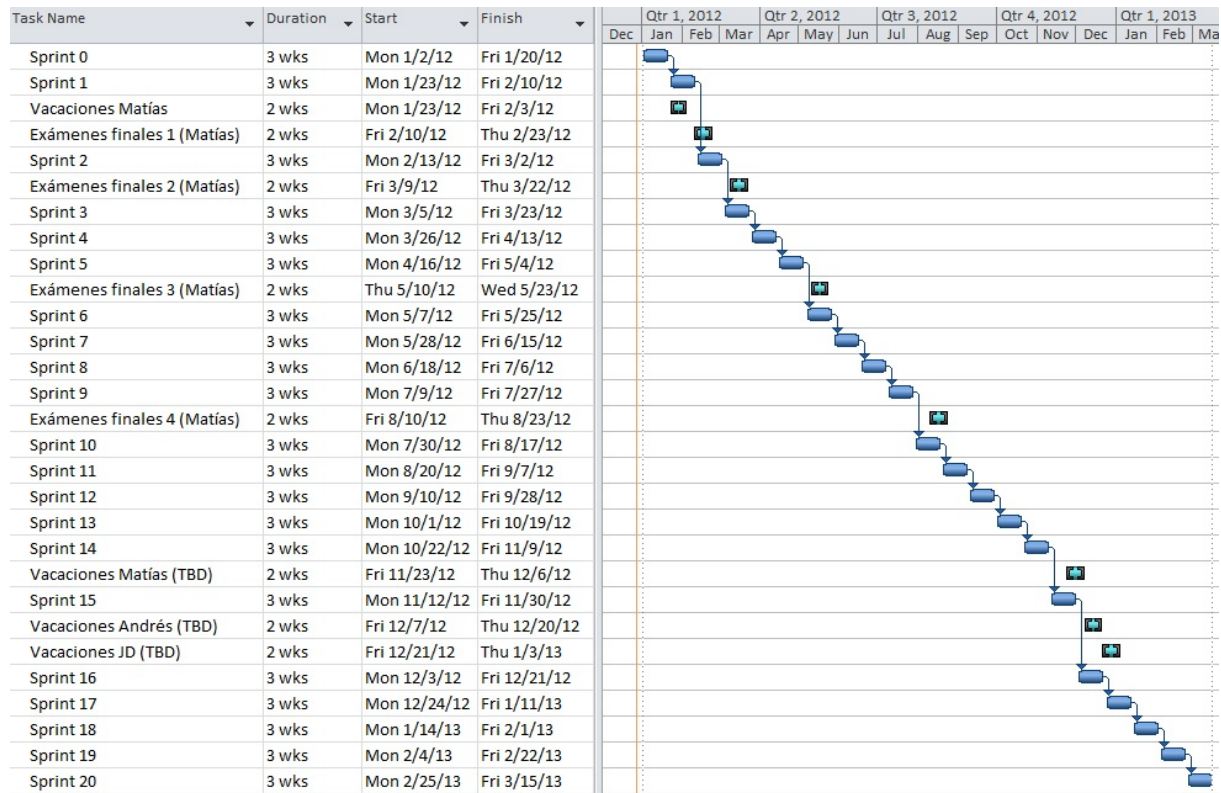
## Planificación

---

Dada que la planificación en cuanto a funcionalidades se define de forma general en el *Product Backlog* y más en detalle al comienzo de cada *sprint*, la planificación existente en este momento es la correspondiente a la estructura de los distintos *sprints* a lo largo del tiempo, junto con la determinación de las tareas para el primer *sprint*.

## Gráfico de Gantt

A continuación se demuestra cómo sería la progresión de los distintos *sprints* hasta la terminación de la ejecución del proyecto.



### Product Backlog (Preliminar)

**Nota:** Este *Product Backlog* es temporal, hasta que se decida cuál será el sistema que mantenga la información del proyecto. Cuando exista esa determinación, los datos se copiarán a dicho sistema y ese será el sistema oficial de registro de *sprints* y tareas.

- Definición de la arquitectura y diseños generales, estrategia técnica de la aplicación (10 hrs)
- Selección de la herramienta a utilizar para repositorio del código fuente (3 hrs)
- Selección de la herramienta a utilizar documentación (3 hrs)
- Selección de la herramienta a utilizar para gestión de tareas (2 hrs)
- Selección del sistema de base de datos a utilizar (3 hrs)
- Selección de la plataforma y UI del sistema (1 hr)
- Determinar sistema de priorización para las tareas incluidas en el *Product Backlog* y las prioridades que tendrán las tareas (10 hrs)
- Migrar *Product Backlog* y demás registros al sistema de tracking del proyecto (5 hrs)
- Creación de los entornos de trabajo y su configuración (Andrés)
- Creación de los entornos de trabajo y su configuración (Juan Diego)
- Creación de los entornos de trabajo y su configuración (Matías)
- Mantener una base de datos de los postulantes (para reemplazar el actual proyecto de Jira “Recruit”)
- Registro de las entrevistas en la base de datos de postulantes
- Permitir búsquedas flexibles e inteligentes sobre la base de datos de postulantes.
- Extracción de datos de los postulantes de CV, LinkedIn, ZonaJobs, Boomeran, BuscoJobs, BaseJobs, etc
- Alta y actualización automática de postulantes desde Email, CV, LinkedIn, ZonaJobs, Boomeran, BuscoJobs, BaseJobs, etc
- Determinación mas o menos inteligente del grado de seniority en postulantes
- Integración con Jira para “Sales Pipeline” con el listado de los postulantes propuestos
- Integrar búsquedas en bases de datos externas como BaseJobs o LinkedIn
- Publicar búsquedas en publicaciones gráficas o sitios como ZonaJobs, Boomeran o BuscoJobs
- Relacionar postulantes con búsquedas realizadas
- Migrar datos actuales (menos de 300 postulantes) del actual proyecto de Jira “Recruit” a nuestra base de datos
- Mantener una base de datos de los empleados (Legajos)
- Integración de los usuarios de JIRA (empleados) con su legajo en nuestra base de datos
- Migrar datos de la planilla actual de legajos a nuestra base de datos (alrededor de 50 ítems)

### Backlog Sprint 0 (Preliminar)

**Nota:** Este *Sprint Backlog* es temporal, que realizemos la *Reunión de Planificación del Sprint*. En ese momento cada integrante comprometerá una cantidad de horas y se analizará cada uno de los ítems incluyendo más detalles.

- Definición de la arquitectura y diseños generales, estrategia técnica de la aplicación (10 hrs)
- Selección de la herramienta a utilizar para repositorio del código fuente (3 hrs)
- Selección de la herramienta a utilizar documentación (3 hrs)
- Selección de la herramienta a utilizar para gestión de tareas (2 hrs)
- Selección del sistema de base de datos a utilizar (3 hrs)
- Selección de la plataforma y UI del sistema (1 hr)
- Determinar sistema de priorización para las tareas incluidas en el *Product Backlog* y las prioridades que tendrán las tareas (10 hrs)
- Migrar *Product Backlog* al sistema de tracking del proyecto (5 hrs)

- Creación de los entornos de trabajo y su configuración (Andrés - 15 hrs)
- Creación de los entornos de trabajo y su configuración (Juan Diego - 15 hrs)
- Creación de los entornos de trabajo y su configuración (Matías - 15 hrs)

Incluyendo 5 horas aproximadas para las reuniones de *planificación*, *revisión* y *retrospectiva*, el total de tiempo estimado para el *Sprint 0*: 87 horas efectivas de trabajo.