
Trabajo Práctico Integrador: primera entrega

Taller de Tecnologías de Producción de Software - Ruby

Cursada 2020



En esta primera entrega práctica vas a comenzar a plantear las bases de una herramienta que construirás a lo largo de la cursada. En las sucesivas entregas vas a ir extendiendo su alcance y modificando incrementalmente lo que tenés hecho hasta el momento para, finalmente, implementar una herramienta más completa a partir de todos los temas que veremos en la materia.

La herramienta

Se te ha encomendado desarrollar una herramienta para llevar un registro personal de notas organizadas de manera que permita acceder fácilmente a la información que éstas contienen.

La herramienta está pensada como un clon simplificado de **Tomboy** implementado, obviamente, en Ruby. Si no conocés la aplicación, Tomboy es una aplicación de código abierto, con licencia LGPL, que corre en entornos de escritorio que permite tomar notas, organizarlas, conectar las notas entre sí y utilizar texto rico para el contenido de las notas, entre otras cosas.

Alcance de la primera entrega

Para completar esta primera entrega, tendrás que diseñar e implementar el modelo de datos básico para que la herramienta provea la siguiente funcionalidad:

- Gestionar las notas con operaciones sencillas (ver apartado **Gestión de notas**).
- Persistir las notas en disco utilizando archivos (ver apartado **Persistencia**).
- Organizar las notas en cuadernos (ver apartado **Organización en cuadernos**).

En esta primera instancia el desarrollo será utilizable mediante la interfaz de línea de comandos (CLI), para lo cual te brindamos una plantilla (o *template*) de proyecto para que utilices como base de tu desarrollo, el cual vas a encontrar en GitHub:

<https://github.com/TTPS-ruby/rn-template>

Podés generar tu proyecto a partir de la plantilla con el botón “*Use this template*” que GitHub provee en el repositorio plantilla. Eso va a permitirte clonar la plantilla en un nuevo repositorio tuyo, y poder trabajar sobre él a partir del código inicial que la cátedra te provee.

Se espera que documentes en algún archivo [README](#) o similar las decisiones de diseño que tomaste, una breve descripción del modelo de datos definido y cualquier otro punto que consideres importante documentar.

Persistencia

Las notas deberán persistirse utilizando archivos de texto plano en el disco. Cada nota se representará con un archivo, que tendrá las siguientes características:

- Su nombre será utilizado como título de la nota, deberá ser único dentro del directorio que lo contenga, y su terminación o extensión será `.rn`.
- Su contenido será de texto plano, y se lo utilizará como el contenido de la nota. *En esta primera etapa sólo utilizaremos notas de texto plano sin formato, pero puede que en el futuro te pidamos que las notas puedan expresarse en Markdown, LaTeX, MediaWiki, o cualquier otro formato que a partir de texto plano permita generar texto rico.*

Una observación respecto de los títulos de las notas:

Tené en cuenta que hay caracteres que no podrán ser parte del título de la nota, porque no está permitido incluirlos en el nombre de un archivo. Esta limitación existirá mientras el proyecto tenga este modo de persistencia utilizando archivos, el cual - tené presente que - en el futuro te pediremos que cambies por otro.

El *cajón de notas* (el lugar en el sistema de archivos donde residen físicamente todas las notas) por defecto será el directorio `.my_rns` dentro del directorio personal ("*Home*") del usuario que ejecute la herramienta, lo cual opcionalmente podés hacer configurable si así lo deseás.

Organización en cuadernos

Las notas deben poder organizarse lógicamente en cuadernos, que serán representados con directorios dentro del cajón de notas. Al igual que con los archivos que representan las notas, el nombre del directorio será el nombre del cuaderno, aplicando a esto las mismas limitaciones que con los títulos de las notas.

Las notas pueden no estar organizadas dentro de un cuaderno, para lo cual existe un cuaderno especial que contendrá las notas que no estén categorizadas dentro de un cuaderno nombrado, al que denominaremos el *cuaderno global*.

Para simplificar la implementación no se permitirá el anidado de cuadernos, es decir que un cuaderno no puede contener a otro(s).

Gestión de notas

La herramienta debe permitir realizar, al menos, las siguientes operaciones:

- Sobre las notas:
 - Creación de una nota.
 - Listado de notas, que puede o no filtrarse por cuaderno.
 - Vista de una nota.
 - Modificación del título una nota.
 - Modificación del contenido una nota.
 - Borrado de una nota.
- Sobre los cuadernos:
 - Listado de los cuadernos de notas.
 - Renombrado de un cuaderno.
 - Borrado de un cuaderno.

Si considerás que sería bueno agregar alguna otra operación sos completamente libre de hacerlo, pero tené en cuenta que es necesario que la implementación que hagas cumpla con las operaciones básicas aquí enumeradas. Es decir: te recomendamos que primero te asegures de cumplir con las operaciones solicitadas por la cátedra y luego, si te queda tiempo antes de la fecha de entrega, implementes las operaciones adicionales.

Podés consultar las operaciones esperadas y los argumentos que recibe cada una de éstas en la plantilla para la entrega.

En general, dejamos a tu criterio cómo implementar la obtención del contenido de las notas. El título, libro y otras opciones se pueden proveer de manera sencilla desde la línea de comandos por tratarse de valores que no se extienden más allá de una línea de texto, pero al intentar expresar el contenido de las notas, que puede fácilmente extenderse más allá de una sola línea de texto, se puede hacer complejo intentar hacer que el contenido sea un argumento más del comando.

Algunas sugerencias al respecto:

- Podés tomar el contenido de la nota como un argumento más (obligando a que se rodee con comillas).
- Similarmente al punto anterior, podés permitir que se provea como un Variadic argument también obligando a que se rodee el contenido con comillas.
- Podés leer el contenido de la nota desde la entrada estándar. Tené en cuenta que deberías tomar los saltos de línea tal cual se ingresen, y tener alguna cadena (¿EOF?) que marque en qué punto tendrías que dejar de leer desde la entrada estándar.
- Cualquier otra idea que tengas que permita ingresar el contenido de la nota de una manera práctica y cómoda, sin complicar demasiado tu implementación.

La decisión que tomes y la implementación deberás dejarlos debidamente documentados en tu entrega.

Modalidad de la entrega

El trabajo, como se ha indicado anteriormente, debe desarrollarse de manera individual. Los trabajos entregados se compararán para encontrar posibles similitudes en su estructura que pudieran provenir de copias entre distintas entregas.

La entrega de esta primera etapa, al igual que las próximas etapas de entrega del Trabajo Práctico Integrador, es de carácter obligatorio pero no así su aprobación. La única etapa del TPI que debe ser aprobada es la última (la *entrega final*). Es decir que en esta etapa podés entregar algo incompleto y con eso será suficiente, pero tené en cuenta que todo lo que no implementes en esta etapa vas a tener que implementarlo más adelante.

Quienes no realicen esta entrega no podrán presentarse a ninguna entrega posterior. Podés no aprobarla, pero es obligatorio que presentes una entrega - aún cuando esté incompleta - en la que se vea que trabajaste.

Tu trabajo tiene que estar versionado en un repositorio Git de GitHub, cuya URL nos entregarás mediante una tarea dedicada a esta primera etapa del TPI en el curso de la plataforma de Cátedras. En caso que el repositorio sea privado, vamos a necesitar que nos brindes acceso al repositorio para poder descargarlo. Si esta es tu situación, por favor consultá antes de la fecha límite de la entrega con el Jefe de Trabajos Prácticos de la materia a qué usuarios de GitHub debés brindarles acceso.

El proyecto debería comenzarse a partir de la plantilla que la cátedra te provee a modo de *template repository* de GitHub (por eso la limitación al hosting para el repositorio Git), que se encuentra disponible en <https://github.com/TTPS-ruby/rn-template>. Si necesitás ayuda con esto, hay un video publicado en la plataforma de Cátedras que muestra cómo trabajar con la plantilla.

La entrega del código se complementará con un coloquio de defensa que se desarrollará según las normativas vigentes al momento del mismo, ya sea de manera virtual mediante la plataforma *Big Blue Button* (BBB) o de manera presencial, si se definiese la habilitación de las clases presenciales en la Facultad.

La fecha límite de entrega del trabajo será comunicada en el curso de la plataforma de Cátedras, y no se admitirán entregas posteriores a esa fecha.