



IIC2115 – Programación como Herramienta para la Ingeniería (II/2019)

Laboratorio 1 - Programación Orientada a Objetos

Objetivos

- Aplicar los contenidos de programación orientada a objetos en la construcción de una herramienta de gestión de proyectos.

Entrega

- **Lenguaje a utilizar:** Python 3.6
- **Lugar:** repositorio privado en GitHub. Recuerde incluir todo en una carpeta de nombre **L01**.
- **Entrega: domingo 1 de septiembre a las 23:59 hrs.**
- **Formato de entrega:** archivo python notebook (**.ipynb**) y archivo python (**.py**) con la solución de este enunciado y modelo entidad relación (**.pdf**). Los archivos deben estar ubicados en la carpeta **L01**. No se debe subir ningún otro archivo a la carpeta. Utilice múltiples celdas de texto y código para facilitar la revisión de su tarea. Los archivos **ipynb** y **py** deben contener la misma solución.
- **Descuentos:** se descontará 0.5 puntos por cada hora de atraso y fracción en la entrega final. Tareas que no cumplan el formato de entrega tendrán un descuento de 0.5 pts.
- **Tareas con errores de sintaxis y/o que generen excepciones serán calificadas con nota 1.0.**
- Si su laboratorio es entregado fuera de plazo, tiene un hasta el **lunes 2 de septiembre a las 23:59 hrs** para responder el formulario de **entregas fuera de plazo** disponible en el Syllabus.
- Las discusiones en las *issues* del Syllabus en GitHub son parte de este enunciado.

Introducción

El problema del cambio climático

En la actualidad, el cambio climático es uno de los problemas más preocupantes que enfrenta la humanidad. A raíz de esta situación, personas e instituciones se encuentran trabajando a favor de la Tierra para generar cambios de conducta en los habitantes. Con el objetivo de mejorar la gestión de proyectos en esta materia, una institución ambiental se encuentra diseñando una plataforma que facilite este proceso. Una vez construida, se espera que facilite la ejecución de medidas para mitigar los efectos del cambio climático. Esta institución ha solicitado ayuda a los programadores del curso Programación como herramienta para la Ingeniería para el desarrollo de esta plataforma. Para agilizar la construcción de este desarrollo, la institución solicitante, a definido una serie de objetivos que se deben cumplir.

Agentes de la plataforma

Se espera que la plataforma sea una herramienta que facilite la gestión de proyectos ambientales, voluntarios e instituciones involucradas. En la siguiente lista se muestran los tipos de proyectos, voluntarios e instituciones que deberá manejar la herramienta.

1. proyecto

- concientización
- ejecución

2. voluntario

- jefe de proyecto
- analista
- obrero

3. institución

- pública
- privada
- social

Proyectos

La plataforma debe ser capaz de crear, editar, visualizar y eliminar proyectos. Estos pueden ser de **concientización**, con el objetivo de educar acerca de temas medioambientales; o de **ejecución**, donde se busca implementar algún proyecto de infraestructura o campaña.

Todos los proyectos deben tener como mínimo: “nombre”, “descripción”, “fecha de inicio”, “estado” (en planificación, en ejecución o finalizado), un **jefe de proyecto**, **analistas**, **obreros** y **logs** (este último campo será explicado más adelante). Además, si el proyecto es de **concientización** debe tener un “tema” e “impacto” (**L**: local, **R**: regional, **N**: nacional o **G**: global). Por otro lado, si es un proyecto de **ejecución**, se debe indicar la “ciudad” y “costo” asociado.

Voluntarios

Respecto al manejo de personas, la plataforma debe ser capaz de crear, visualizar y eliminar voluntarios. Un voluntario puede ser **jefe de proyecto**, **analista** u **obrero**. Como mínimo, un voluntario debe tener: “nombre”, “edad”, “fecha ingreso” y “próxima acción”. Esta última corresponde a la siguiente actividad que debe realizar ese voluntario dentro de un proyecto (esto será explicado más adelante).

Un **jefe de proyecto** puede tener a cargo uno o más proyectos, así como **analistas** y **obreros** pueden participar en más de un proyecto. Los **analistas** poseen un grado de “especialización” inicial o avanzada. **Analistas** y **obreros** están bajo la jefatura de un jefe de proyecto.

Instituciones

Cada proyecto puede tener relación con un máximo de dos **instituciones**, estas pueden brindar “apoyo” económico, técnico o de gestión. De este modo, se debe poder crear, visualizar y eliminar instituciones. Toda institución posee un “nombre”. Si se trata de un apoyo económico, estas solo pueden ser instituciones **públicas** o **privadas**.

A modo de resumen, en la herramienta se cumple que:

- existen proyectos administrados por jefes de proyecto y con la posible colaboración de hasta dos instituciones.
- cada jefe de proyecto, en cada proyecto posee un grupo de personas a cargo. Estos pueden ser analistas y/u obreros.

Lógica de funcionamiento

La plataforma debe permitir la creación de voluntarios, proyectos e instituciones pidiendo los campos necesarios para crearlo (dependiendo de qué objeto sea). Los proyectos que se crean en estado de “en planificación” requieren cómo mínimo solo un nombre. Respecto a los proyectos en general, estos pueden ser editados en cualquiera de sus campos. En este caso, es importante ser consistente de los cambios de valor en el resto de variables. De igual forma, la eliminación de proyectos, voluntarios e instituciones debe ser “inteligente”, es decir, si se elimina un voluntario que es jefe de proyecto, este no puede quedar sin jefe; o si se elimina un proyecto, este cambio debe afectar a los voluntarios, etc.

La visualización de proyectos, instituciones o voluntarios debe permitir ver los campos “útiles” (a su criterio) del objeto a visualizar.

Próxima acción

El campo *logs* de cada proyecto lleva un registro de qué actividades se han realizado en cada uno de los proyectos. En este sentido, los voluntarios deben implementar la funcionalidad “acción”. Al ser utilizada por algún voluntario, este realiza la acción definida en su campo “próxima acción”. Una vez ejecutada, esta se registra en los *logs* del proyecto. Si esta funcionalidad es ejecutada por un jefe de proyecto, entonces puede asignar el campo “próxima acción” en alguno de sus voluntarios a cargo para algún proyecto e igualmente se registra en los *logs* del proyecto correspondiente.

Misiones para la herramienta

Para completar requerimiento solicitado por la institución ambiental, usted deberá crear una plataforma que implemente la lógica presentada. Para ello deberá completar una serie de misiones que le guiarán a cumplir su cometido.

- M1. Antes que nada, debes dejar el código de lado. Tu primera misión será definir correctamente tu modelo de entidad relación. Lee atentamente lo que debes crear y define el modelo adecuado para relacionar proyectos, voluntarios e instituciones. Debes utilizar correctamente los conceptos de herencia y polimorfismo, estos serán útiles para definir tu modelo. Para cumplir esta misión, debes entregar junto con tus archivos de laboratorio, un PDF (una sola plana) con tu modelo de entidad relación. (2.1 puntos)
- M2. Una vez definido el modelo de entidad relación, ya es hora de programar. Debes crear los objetos definidos en la primera misión. (0.6 puntos)

- M3. Debes crear una visualización de bienvenida. Como mínimo, esta debe incluir las opciones: ingresar un voluntario, ingresar una institución, crear un proyecto, gestionar un proyecto, visualizaciones, eliminar, salir. Basta con que construyas una vista que permita seleccionar alguna de estas opciones. (0.1 puntos)
- M4. Ingresar un voluntario. En esta misión deberás implementar la funcionalidad que permite ingresar un voluntario. Para ello, debes solicitar los campos indicados anteriormente. (0.2 puntos)
- M5. Ingresar una institución. En esta misión deberás implementar la funcionalidad que permite ingresar una institución. (0.2 puntos)
- M6. Crear un proyecto. En esta misión deberás implementar la funcionalidad que permita crear un proyecto. Recuerda que dependiendo el estado del proyecto se solicitan campos diferentes. (0.3 puntos)
- M7. Gestionar un proyecto - edición. Esta funcionalidad debe permitir la edición de cualquiera de los campos. La edición de campos debe ser consistente y debe propagar el cambio al resto de las variables que se relacionen con el cambio. (0.3 puntos)
- M8. Visualizaciones. En funcionalidad debe permitir la visualización de proyectos, voluntarios o instituciones. Puedes utilizar el campo nombre para ayudar al usuario en la selección de qué desea visualizar. (0.3 puntos)
- M9. Eliminar. Esta funcionalidad debe permitir al usuario eliminar proyectos, instituciones o voluntarios. Recuerda que se debe respetar la consistencia de los cambios que resulten. (0.3 puntos)
- M10. Gestionar un proyecto - acciones. Esta funcionalidad debe permitir el manejo de acciones definidas en el enunciado. (0.4 puntos)
- M11. Gestionar un proyecto - visualización de *logs*. Esta funcionalidad debe permitir visualizar la lista de *logs* para algún proyecto seleccionado. Estos deben estar ordenados desde el más antiguo al más nuevo a la hora de visualizarse. (0.2 puntos)
- M12. Funciones extra - voluntarios estrella. Debes implementar una nueva opción en la bienvenida con el nombre de “funciones extra”. Una herramienta dentro de esta opción y que debes implementar para completar esta misión es “voluntarios estrella”. En esta funcionalidad debes listar a los 3 trabajadores que tengan la mayor participación en proyectos (indicado en cuantos proyecto participa junto a su nombre). (0.4 puntos)

M13. Funciones extra - filtro de proyectos. Esta opción debe permitir ver una lista de los proyectos según categoría (concientización o ejecución) o su antigüedad en la fecha de inicio (*i.e.* proyectos que iniciaron hace más de 2 meses). (0.4 puntos)

M14. Poblado de datos. En esta misión deberás implementar un método, que se ejecute al inicio, y que importe 3 proyectos, 3 jefes de proyecto, 5 analistas, 5 obreros y 3 instituciones. Esta misión es importante dado que será utilizada para revisar tu tarea. (0.2)

* para el manejo de fechas, utilice la librería *datetime*

Corrección

Para la corrección de este laboratorio, se revisará el modelo de entidad relación construido y la implementación de la distintas misiones. Es importante que al enviar su tarea, la ejecución contemple que ya existen 3 proyectos, 3 jefes de proyecto, 5 analistas, 5 obreros y 3 instituciones.

Política de Integridad Académica

“Como miembro de la comunidad de la Pontificia Universidad Católica de Chile me comprometo a respetar los principios y normativas que la rigen. Asimismo, prometo actuar con rectitud y honestidad en las relaciones con los demás integrantes de la comunidad y en la realización de todo trabajo, particularmente en aquellas actividades vinculadas a la docencia, el aprendizaje y la creación, difusión y transferencia del conocimiento. Además, velaré por la integridad de las personas y cuidaré los bienes de la Universidad.”

En particular, se espera que mantengan altos estándares de honestidad académica. Cualquier acto deshonesto o fraude académico está prohibido; los alumnos que incurran en este tipo de acciones se exponen a un procedimiento sumario. Ejemplos de actos deshonestos son la copia, el uso de material o equipos no permitidos en las evaluaciones, el plagio, o la falsificación de identidad, entre otros. Específicamente, para los cursos del Departamento de Ciencia de la Computación, rige obligatoriamente la siguiente política de integridad académica en relación a copia y plagio: Todo trabajo presentado por un alumno (grupo) para los efectos de la evaluación de un curso debe ser hecho individualmente por el alumno (grupo), sin apoyo en material de terceros. Si un alumno (grupo) copia un trabajo, se le calificará con nota 1.0 en dicha evaluación y dependiendo de la gravedad de sus acciones podrá tener un 1.0 en todo ese ítem de evaluaciones o un 1.1 en el curso. Además, los antecedentes serán enviados a la Dirección de Docencia de la Escuela de Ingeniería para evaluar posteriores sanciones en conjunto con la Universidad, las que pueden incluir un procedimiento sumario. Por “copia” o “plagio” se entiende incluir en el trabajo presentado como propio, partes desarrolladas por otra persona. Está permitido usar material disponible públicamente, por ejemplo, libros o contenidos tomados de Internet, siempre y cuando se incluya la cita correspondiente.