



GOBIERNO DE LA CIUDAD DE BUENOS AIRES

Ministerio de Educación

Dirección de Formación Técnico Superior

Instituto de Formación Técnico Superior N° 18

Mansilla 3643 - C1425BBW - Capital Federal

CARRERA de TÉCNICO SUPERIOR en DESARROLLO DE SOFTWARE

Res. 1148-MEGC-2007

Trabajo Práctico Final

Materia: Desarrollo de Sistemas Orientados a Objetos

Curso: 1er Año

Profesor: Eduardo Iberti

Ciclo Lectivo: 2do Cuatrimestre 2022

Fecha de entrega: jueves 24 de noviembre

El trabajo se puede realizar en forma individual o en grupos de hasta tres alumnos.

Tema: Sistema de gestión de obras públicas con manejo de POO, importación de datasets desde un archivo csv y persistencia de objetos en una base de datos SQLite.

Enunciado y requerimientos funcionales

Se requiere desarrollar un software en Python para gestionar las obras urbanas de la Ciudad de Buenos Aires, respetando el modelo de Diagrama de Clases UML anexo.

El modelo de negocio debe complementarse con los siguientes requerimientos funcionales:

1. En primer lugar se deben importar los datos de las obras ya registradas en el archivo csv <https://cdn.buenosaires.gob.ar/datosabiertos/datasets/secretaria-general-y-relaciones-internacionales/ba-obras/observatorio-de-obras-urbanas.csv> a las tablas correspondientes de la base de datos relacional SQLite. Esto debe realizarse invocando al método de clase importar_csv() de la clase abstracta GestionarDAO, desde el método de clase main() de la clase abstracta Main.
2. También desde el método de clase main() de la clase abstracta Main, deberán crearse al menos tres nuevas instancias de obras, dos de ellas deberán finalizar en tiempo y forma y la restante deberá ser rescindida. Las tres obras deberán ser de diferente tipo de contratación, y deben corresponderse con alguno de los tipos de contratación ya existentes en la base de datos importada.



3. La creación de instancias de Obras se debe realizar en el método de clase `nueva_obra()` de la clase abstracta `GestionarModelo`, el cual debe recibir como parámetros los valores de los atributos que se inicializarán con la instancia. A continuación, el objeto instanciado debe agregarse a la lista del atributo de clase privado "listado_obras".
4. Al momento de crearse las instancias de Obra, deben respetarse los siguientes requisitos:
 - a. Su etapa inicialmente debe tener el valor "Proyecto". Si dicha etapa no existe en la BD, deberá crear la instancia desde el método de clase `nueva_etapa()` y luego deberá insertar el nuevo registro en la tabla correspondiente.
 - b. Deben asignarse valores para los siguientes atributos: entorno, nombre, etapa, tipo_obra, area_responsable, descripción, monto_contrato, barrio, dirección, plazo_meses, beneficiarios. Los valores de los atributos tipo_obra, area_responsable y barrio deben ser alguno de los existentes en la base de datos.
5. A continuación, debe iniciarse la licitación/contratación, para ello deberá ejecutar el método `iniciar_contratacion()`, asignando el `TipoContratacion` (debe ser un valor existente en la BD) y el `nro_contratacion`.
6. Para adjudicar la obra a una empresa, deberá ejecutar el método `adjudicar_obra()` y asignarle la Empresa (puede ser una empresa existente o una nueva) y el `nro_expediente`.
7. Para indicar el inicio de la obra, deberá ejecutar el método `iniciar_obra()`, y asignarle valores a los siguientes atributos: destacada, fecha_inicio, fecha_fin_inicial, fuente_financiamiento (debe ser un valor existente en la BD) y `mano_obra`.
8. Para registrar avances de la obra, deberá ejecutar el método `actualizar_porcentaje_avance()` y actualizar el valor del atributo `porcentaje_avance`.
9. Para incrementar el plazo de la obra, deberá ejecutar el método `incrementar_plazo()` y actualizar el valor del atributo `plazo_meses`.
10. Para agregar imágenes de la obra, deberá ejecutar el método `agregar_imagenes()` creando instancias `Imagen` con el nombre de la imagen y agregarlas al atributo `imagenes` de tipo lista.
11. Para incrementar la cantidad de mano de obra, deberá ejecutar el método `incrementar_mano_obra()` y actualizar el valor del atributo `mano_obra`.



12. Para indicar la finalización de una obra, deberá ejecutar el método finalizar_obra() y actualizar el valor del atributo etapa a “Finalizada” y del atributo porcentaje_avance a “100”.
13. Para indicar la rescisión de una obra, deberá ejecutar el método rescindir_obra() y actualizar el valor del atributo etapa a “Rescindida”.
14. Luego de finalizada o rescindida la obra, debe persistirse en la base de datos el objeto Obra con todos sus atributos, en las tablas que correspondan.
15. Para finalizar, se requiere obtener desde la base de datos (con sentencias SQL) y mostrar por consola los siguientes valores. Para ello deberá agregar los métodos que considere necesarios en la clase Obra_DAO:
 - a. Listado de todas las áreas responsables.
 - b. Listado de todos los tipos de obra.
 - c. Cantidad de obras que se encuentran en cada etapa.
 - d. Cantidad de obras por tipo de obra.
 - e. Listado de todos los barrios pertenecientes a las comunas 1, 2 y 3.
 - f. Cantidad de obras “Finalizadas” en la comuna 1.
 - g. Cantidad de obras “Finalizadas” en un plazo menor o igual a 24 meses.

Aclaraciones:

1. Todas las clases del modelo de negocio (módulo “model”), deben tener propiedades para sus atributos privados y un método __init__() para inicializar los valores de los atributos que correspondan.
2. En todas las clases abstractas del módulo “util”, sus métodos deben ser métodos de clase y sus atributos (en caso que corresponda) también serán atributos de clase.
3. En relación al módulo “dao”, se deberá completar la funcionalidad del método modificar_registro() en todas las clases que implementan la interface DAO. También podrá agregar nuevos métodos en algunas de las clases DAO, si lo considera necesario.