

Módulo: Programador

Proyecto ABP

Introducción al proyecto

El **proyecto final** será compuesto de dos etapas, una correspondiente a la **evidencia 2** cuya consigna es detallada en este documento, y una siguiente etapa que será parte de la **evidencia 3** (consigna aún no publicada). La resolución de cada evidencia es especificada en detalle en el apartado de “Entregables” que posee cada consigna de evidencia. El trabajo desde evidencia 2 en adelante es grupal, si bien pueden dividirse tareas, cada uno debe saber de la totalidad de lo abordado en cada consigna.

Algunas fechas importantes:

Evidencia 2 tiene como fecha de entrega el viernes 16/5 a las 23:59hs.

Evidencia 3 tiene como fecha de entrega el viernes 30/5 a las 23:59hs. Su consigna aún no es presentada.

Una vez presentada la consigna de la Evidencia 3, con ella también se compartirá el modelo del documento que deben elaborar, y que será parte de la entrega final.

El documento del proyecto final aún no tiene fecha establecida de entrega.

La defensa oral del Proyecto ABP será el día viernes 13/6 en los horarios de sincrónico previa distribución de horarios (a confirmar).

Veamos ahora de que se trata la temática del proyecto:

Situación Profesional

El presente proyecto final surge como respuesta a una situación profesional concreta planteada por la cooperativa agrícola local AgroTech Coop, la cual busca implementar una solución digital que le permita monitorear de manera eficiente las condiciones clave de sus parcelas de cultivo. Frente a esta necesidad, deberán asumir el rol de programadores encargados de desarrollar un sistema funcional y

escalable que registre datos provenientes de sensores IoT y los presente de manera comprensible para asistir la toma de decisiones productivas.

Una cooperativa agrícola local, "AgroTech Coop", busca implementar un sistema básico para monitorear las condiciones clave de sus parcelas de cultivo utilizando sensores IoT (Internet de las Cosas).

Han contratado a los estudiantes para desarrollar un prototipo de software que permita registrar datos de sensores y visualizar información relevante para la toma de decisiones en el manejo de los cultivos.

Este sistema permitirá a los técnicos agrícolas registrar parcelas, tipos de sensores (humedad del suelo, temperatura ambiente, pH del suelo) y los datos recolectados por estos sensores.

La cooperativa busca una solución funcional y escalable, priorizando una correcta estructuración de los datos para futuras ampliaciones. El proyecto deberá sentar las bases para un manejo eficiente de la información recolectada.

El proyecto estará basado en una aplicación de escritorio (consola) python que utilizará base de datos relacional y que integrará los saberes de todo el Módulo programador. La aplicación se presentará al usuario mediante un menú de opciones que permitirá resolver lo requerido por la cooperativa agrícola.

Objetivo Principal

Crear un prototipo de sistema que permita:

1. Registrar las distintas parcelas de cultivo de la cooperativa.
2. Registrar los tipos de sensores disponibles y los sensores específicos instalados en cada parcela.
3. Registrar las mediciones enviadas por los sensores (ej. nivel de humedad, temperatura, etc.).
4. Consultar las últimas mediciones registradas por parcela o por tipo de sensor.

5. Diseñar una base de datos relacional que soporte la información del sistema.
6. Formular consultas SQL para la creación de la estructura de la base de datos y la manipulación de datos básicos.

Requerimientos generales funcionales del sistema

- **Gestión de Parcelas:** Permitir registrar nuevas parcelas con detalles como ubicación, tamaño y tipo de cultivo principal.
- **Gestión de Sensores:** Permitir registrar tipos de sensores (ej. "Termómetro Digital Agrícola Modelo X", "Sensor de Humedad Suelo Modelo Y") y los sensores individuales instalados, asociándolos a una parcela.
- **Registro de Mediciones:** Permitir registrar los datos enviados por cada sensor, incluyendo el valor medido, la unidad (ej. °C, %, pH) y la fecha/hora de la medición.
- **Consulta de Datos:** Permitir visualizar las parcelas registradas, los sensores por parcela y las últimas mediciones registradas.

Requerimientos no funcionales

- **Modularidad.** El código fuente debe estar estructurado en módulos y funciones independientes para facilitar la escalabilidad, mantenimiento y comprensión.
- **Legibilidad.** El código fuente debe ser claro, con variables, funciones y módulos.
- **Eficiencia.** El programa debe optimizar el uso de recursos del sistema como memoria y CPU.
- **Documentación.** Incluir documentación adecuada y completa que explique cómo instalar, configurar y ejecutar el programa. Se valorará que además se detalle una explicación de uso del programa.

- **Usabilidad.** La interfaz de consola debe ser intuitiva y permitir al usuario interactuar fácilmente con el programa, esto es, la navegación debe ser pensada en el usuario.

Evidencia de aprendizaje N° 2 - Consigna

A partir de la **Evidencia de Aprendizaje N.º 2**, se inicia la construcción de un primer prototipo de aplicación de consola en Python, estructurado bajo el paradigma imperativo y utilizando estructuras básicas de programación. Este prototipo tiene como finalidad establecer las bases funcionales del sistema, ofreciendo un menú interactivo que simula el funcionamiento deseado. Paralelamente, se diseña un modelo relacional de base de datos que organiza de manera coherente toda la información relevante para la cooperativa.

Programación

Desarrollar un primer prototipo de la aplicación de consola, en un único archivo python (`main.py`), utilizando solo estructuras básicas de programación: estructuras secuenciales, condicionales e iterativas.

Contenidos de Programación Aplicados:

- Paradigma Imperativo: programas secuenciales.
- Estructuras de control: condicionales (`if/else`) y ciclos (`for, while`).
- Contadores y Acumuladores.
- (*opcional*) Control de versiones: Git y GitHub (creación de repositorio, push inicial).

Alcance funcional en esta evidencia:

- Implementar un menú de opciones interactivo, por ejemplo (NO necesariamente debe ser así):
 - Gestionar Parcela (si ingresa a Gestionar Parcela tendrá un submenú con opciones como ...)
 - Ver Parcela
 - Agregar Parcela
 - Modificar Parcela
 - Eliminar Parcela
 - Gestionar Sensores

- *opciones que se consideren*
 - Registrar Mediciones
 - *opciones que se consideren*
 - Consultar Datos
 - *opciones que se consideren*
 - Salir del sistema.
- Cada opción mostrará un mensaje representativo. (No debe haber aún funciones ni módulos externos). Por ejemplo, una navegación posible al ejecutar `main.py` podría mostrar mediante prints lo siguiente:
 - *Bienvenidos a el Monitoreo de "AgroTech Coop", cooperativa agrícola local (y el menú de opciones)*
 - *Ingreso a Gestionar Parcela*
 - *eligio la opción 2 agregar parcela*
 - *ingreso los siguientes datos para la nueva parcela:*
 - *Se guardó la parcela*
 - *Eligio ver parcelas*
 - *Eligió salir (y sale de la aplicación)*

Estructura mínima requerida:

- Carpeta de proyecto con:
 - *(opcional)* Repositorio Git inicializado y vinculado a GitHub.
 - Archivo `README.md` que contenga:
 - Nombre del proyecto.
 - Breve descripción.
 - Integrantes del grupo (nombre, apellido y DNI).
 - Instrucciones básicas de ejecución.
 - Archivo `main.py` que contenga:
 - El menú de opciones.
 - Control de flujo de acciones a realizar según la opción seleccionada (sin modularización).

Base de Datos

1. Identificar las entidades principales necesarias para almacenar la información del sistema de monitoreo agrícola (ej. Parcelas, TiposDeSensor, Sensores, Mediciones).

2. Definir los atributos relevantes para cada entidad, especificando el tipo de dato adecuado para cada uno (ej. texto, número entero, número decimal, fecha/hora).
3. Establecer las relaciones entre las entidades identificadas (uno a muchos, muchos a muchos), especificando las claves primarias y foráneas necesarias.
4. Crear el **modelo relacional** completo, representando gráficamente las tablas, sus atributos y las relaciones entre ellas. Utilizar herramientas como draw.io, lucidchart u otras similares.

Ética y Ejercicio Profesional

1. Elaborar un informe reflexivo y normativo que analice las implicancias éticas y legales del rol profesional del programador en el contexto de un desarrollo tecnológico aplicado al agro, como el sistema solicitado por la cooperativa AgroTech Coop.

El informe deberá considerar:

- La relación laboral que se establece en el desarrollo del software solicitado por la cooperativa, encuadrada en la Ley N.º 20.744 (Contrato de Trabajo): derechos y obligaciones del profesional contratado, y modalidades posibles del vínculo laboral.
- La importancia de las asociaciones sindicales en el sector tecnológico y su rol en procesos de negociación colectiva (Ley N.º 23.551 y N.º 14.250).
- Una reflexión personal sobre cómo se vincula el perfil del programador con la responsabilidad profesional y social, al diseñar una solución que influye en las decisiones productivas y ambientales de una cooperativa agrícola.

Entregables

1. Programación I.

Link al repositorio GitHub con el proyecto (o entregar un archivo compactado con el nombre del grupo (ejemplo: **grupo5.zip**) y que contenga los archivos **README.md** y **main.py**).

Documento que contenga el código **main.py** y las observaciones sobre el desarrollo de las consignas de programación que consideren oportuno agregar.

2. Base de Datos.

Documento que contenga el diagrama del modelo relacional generado.

3. Ética y ejercicio profesional.

Documento que contenga el informe solicitado con referencias normativas claras y una sección final de conclusiones personales vinculadas al caso AgroTech Coop.

4. Nos adelantamos al Informe Final. Inicio de la elaboración del Documento Final del Proyecto.

Siguiendo la "Estructura para el diseño de Proyectos ABP" provista (Documento que pronto será subido en el mosaico Proyecto Final). Para esta entrega, un avance sugerido del informe es incluir los siguientes puntos **completados o en borrador inicial**:

- **Problemáticas/Necesidades:** Identificación y descripción inicial de la problemática a abordar. La misma surge de la situación profesional y el enunciado de la evidencia de aprendizaje 2.
- **Fundamentación:** Un borrador inicial explicando la importancia de abordar esta problemática/necesidad identificada
- **Producto Final:** Una descripción inicial de la parte del producto final que están construyendo en esta etapa.

Confeccionar un solo documento en formato PDF.

Añadir Carátula con datos de identificación de cada estudiante del grupo.

Formato: Letra Arial Tamaño 11, interlineado 1. Texto justificado. Márgenes: Superior a 4 cm., Inferior a 4 cm., Izquierdo a 4 cm., Derecho a 2 1/2 cm.

Evidencia de aprendizaje N° 3

Se habilitará la consigna más adelante.

Documento del Proyecto Final ABP

Se habilitará más adelante el documento modelo con todos los puntos a considerar.