

Python - CRM (Customer Relationship Management)

Objetivos

- Afianzar y profundizar los conocimientos de programación en general y de programación en Python en particular.
- Profundizar conocimientos en los softwares **ERP (Enterprise Resource Planning)**, y concretamente en el módulo CRM (**Customer Relationship Management**) de los mismos.

Consideraciones iniciales:

1. Para la interfaz gráfica debes utilizar la librería PyQt (que es una adaptación de la biblioteca gráfica Qt para el lenguaje de programación Python).
2. Para implementar la base de datos puedes utilizar SQLite, MySQL, PostgreSQL, etc, aquella con la que más cómodo te sientas trabajando y/o te permita una mayor funcionalidad según tus necesidades.

Requisitos no funcionales de la aplicación:

1. Dentro de la carpeta del proyecto, tanto las BBDDs como los iconos, imágenes, etc, que se vayan a utilizar en la aplicación, deberán estar en carpetas diferentes.
2. Ventana principal de la app de tamaño mínimo 20x30 cm (más o menos como una pantalla de 12").
3. Diseños de ventana, estilos de fuente, colores, estilo de menús, botones y diferentes elementos atendiendo a criterios de usabilidad.

Requisitos funcionales de la aplicación:

Deberás implementar un CRM (Customer Relationship Management) en Python con las siguientes funciones y características:

1. Inicialmente, el programa presentará una pantalla para loguearse/darse de alta en la aplicación. Aquí se pedirá como mínimo un nombre de empresa, correo electrónico y una contraseña (habrá que confirmar la contraseña). Una vez
2. Respecto a la base de datos, la app debe crear una nueva base de datos para cada empresa que queramos registrar en nuestro CRM, para ello bastará con loguearse como una nueva empresa. Dentro de la base de datos de cada empresa, debe existir la tabla IDENTIFICACION con los campos NOMBRE_EMPRESA y PASSWORD y MAIL, que serán aquellos que se comprobarán al loguearse.
 - 2.1. La base de datos tendrá el mismo nombre que la empresa)
 - 2.2. Una vez accedemos a lavase de datos, nos debe aparecer una pantalla de bienvenida a nuestra app y el menú de opciones que podemos realizar. Dentro del menú tendremos, como mínimo, los siguientes ítems: PIPELINE, EMPRESA, CLIENTES, OPORTUNIDADES, PRESUPUESTOS e INVENTARIO.

- 2.3. Nuestro PIPELINE debe constar de al menos 4 fases para las oportunidades de ventas. Estas fases son: NUEVO, CALIFICADO, PROPUESTA, GANADO, (puedes cambiar sus nombres o el número de ellas si quieres), y en ellas nos aparezcan las distintas oportunidades que se hayan generado en la empresa hasta ese momento en su estado correspondiente. Además, desde el pipeline podremos cambiar la fase en la que se encuentran nuestras oportunidades de venta. También deberemos de poder ir a la ventana de OPORTUNIDADES desde la ventana de PIPELINE.
3. Desde EMPRESA, como mínimo podremos:
- 3.1. Visualizar los datos de nuestra empresa.
 - 3.2. Cambiar los datos de nuestra empresa excepto el nombre (si no, no tendría sentido el nombre de la base de datos).
4. Desde CLIENTES podremos:
- 4.1. Crear nuevos clientes, que serán clientes potenciales para nuestra empresa, pudiéndose tratar también de otras empresas. Los campos mínimos para los clientes serán: ID_CLIENTE y/o CIF, NOMBRE, DIRECCIÓN (CALLE, CODIGO POSTAL, CIUDAD, PAIS), TELÉFONO, PERSONA DE CONTACTO, CORREO ELECTRÓNICO.
 - 4.2. Editar y cambiar datos de nuestros clientes, salvo el campo ID_CLIENTE o CIF (dependiendo del que hayas elegido como primary key).
 - 4.3. Consultar nuestros clientes.
5. Desde OPORTUNIDADES podremos:
- 5.1. Crear nuevas oportunidades con los campos mínimos: ID_OPORTUNIDAD, NOMBRE_OPORTUNIDAD, CLIENTE, FECHA, PRESUPUESTO, INGRESO ESPERADO y ESTADO. **Los campos CLIENTE y PRESUPUESTO deberán poder ser elegidos dentro de los clientes y presupuestos ya creados**, si no, deberemos de poder acceder a las ventanas de CLIENTES y/o PRESUPUESTOS desde OPORTUNIDADES.
 - 5.2. Consultar las oportunidades.
 - 5.3. Editar las oportunidades para posibles cambios.

NOTA: Una oportunidad solo puede contener un presupuesto.

6. Desde PRESUPUESTOS podremos:
- 6.1. Crear un nuevo presupuesto, con los campos mínimos ID_PRESUPUESTO, NOMBRE, FECHA_CREACION, FECHA_EXPIRACION, PRODUCTOS_PRESUPUESTO, SUBTOTAL_PRESUPUESTO (sin IVA), TOTAL_PRESUPUESTO (con IVA).
 - 6.2. Al crear los presupuestos, el campo CLIENTE deberá de poder ser elegido de entre los clientes ya creados, o bien, en el caso de ser un cliente nuevo, deberá ser posible acceder a la ventana de creación de clientes desde la ventana de presupuestos.
 - 6.3. Un presupuesto podrá contener uno o varios productos, que deberán poder ser elegidos entre los ya creados o bien, poder acceder a la creación de los mismos desde la ventana de presupuestos.

NOTA: Los diferentes productos podrán ser tanto añadidos como quitados antes de la formalización de un nuevo presupuesto. En el caso de elegir varias veces un producto, éste solo debe aparecer una vez, modificándose la cantidad del mismo.

- 6.4. Consultar y editar los presupuestos ya creados. En el caso de consulta, al realizarse ésta, deberán de aparecer en la misma todos los productos de que consta ese presupuesto, además de los demás campos que lo conforman.

- 6.5. El total del presupuesto dependerá de los productos del mismo y del tipo de IVA que se aplique a cada producto.
7. Desde INVENTARIO podremos:
- 7.1. Comprar nuevos productos. Cada producto consta como mínimo de los campos ID_PRODUCTO, PROVEEDOR_PRODUCTO, NOMBRE_PRODUCTO, DESCRIPCION_PRODUCTO, IVA_PRODUCTO, PRECIO, STOCK_PRODUCTO. El campo STOCK_PRODUCTO deberá irse actualizando tanto si compramos productos como si vendemos u ofertamos producto.
 - 7.2. Editar producto para posibles cambios.
 - 7.3. Consultar y editar los productos ya creados.

NOTA: Esta parte del inventario no formaría parte de un CRM tal y como lo conocemos, ya que formaría parte del módulo de inventario y/o del módulo de compras, pero se nos hace necesaria para el funcionamiento de la aplicación.

Presentación de la tarea

1. Debes realizar un informe en PDF de la práctica en el que deberás recoger todos los aspectos del programa, como son las distintas fases del mismo (**análisis, diseño y desarrollo**). En estas fases explicarás con detalle las distintas soluciones adoptadas (creación de la interfaz gráfica, creación de la BBDD y su modelo relacional, etc). **Respecto a la programación, es indispensable utilizar clases y objetos y tener el código comentado.** En el mismo informe, en un anexo, debes recoger un pequeño manual de usuario de tu programa.
2. Debes realizar una defensa en clase de tu programa, por lo que deberás preparar una presentación y demostración del mismo de 20 minutos de duración como máximo. Es aconsejable que para esta tarea te apoyes en una presentación PowerPoint o similar.

Calificación

- Para que la práctica esté aprobada (un 5 / 10) deberán de funcionar todos los aspectos anteriormente citados. A partir de ahí, la nota se complementará con el informe, la presentación y las mejoras introducidas en la misma. **Si el programa no funcionase en alguno de los aspectos básicos anteriormente citados, la práctica se consideraría SUSPENSA y el trimestre también.**
- La nota de esta actividad corresponderá al 95 % de la evaluación, dejando un 5% a la evaluación de las distintas competencias transversales y/o blandas.

Posibles mejoras

- Menús u otros elementos que se muestren u oculten en función de donde tengamos posicionado el ratón etc.
- Crear diferentes usuarios con diferentes roles para el CRM.
- Al realizar un pedido u oferta, el programa deberá ser capaz de generar un documento de texto en formato PDF con el pedido u oferta realizada para poder enviarlo a los clientes.

- Posibilidad de mandar emails a contactos existentes y a otros posibles/futuros clientes, con las diferentes oportunidades de venta creadas.
- Posibilidad de generar gráficos estadísticos de nuestras oportunidades, productos, clientes, etc.
- Introducir un logo en la base de datos de las empresas para que éste aparezca en el CRM de las mismas.
- Etc.

Documentación

- ✓ PyQt5: Interfaces gráficas en Python 3, Alberto Cuevas Álvarez.
- ✓ <https://www.pythonguis.com/pyqt5-tutorial/>
- ✓ Primeros pasos en PyQt 5 y Qt Designer <https://medium.com/@hektorprofe/primeros-pasos-en-pyqt-5-y-qt-designer-programas-gr%C3%A1ficos-con-python-6161fba46060>
- ✓ Información de matplotlib: Empleada para generar gráficos en 2D y 3D. <https://matplotlib.org/> . Observa la sección "Getting started" que es bastante ilustrativa para hacer cosillas básicas.
- ✓ Mayavi: Biblioteca de visualización científica 3D en Python. <https://docs.enthought.com/mayavi/mayavi/>