**Antonio Jiménez Sevilla**

**TareaSGE05**

|  |
| --- |
| **Enunciado.** |
| En esta unidad hemos aprendido cómo crear nuevos componentes utilizando sentencias del lenguaje propio del sistema ERP-CRM. Hemos creado componentes de manipulación de datos mediante módulos que crean a su vez tablas en la base de datos. También hemos añadido módulos al sistema y comprobado que funcionan. Además, hemos conocido herramientas adicionales para la creación de formularios e informes.  Ahora es el momento que pongamos en práctica estos conocimientos. La tarea consiste en crear un componente o módulo que gestione una **agenda telefónica** con las siguientes características:  **Modelo y Controlador.**   * Objeto en la aplicación llamado agenda con, al menos, dos campos: Nombre y Teléfono. * Tabla en la base de datos con los datos del objeto.   **Vista.**   * Menú en la aplicación que enlace al objeto. * Vista formulario con los datos del objeto. * Vista árbol con los datos del objeto.   Además del módulo deberás escribir también un informe con todas las consideraciones oportunas que se necesiten para entender cómo has realizado la tarea. |

**Ejercicio**

Interfaz de usuario gráfica, Texto, Aplicación, Chat o mensaje de texto

Descripción generada automáticamenteComo IDE para crear el módulo se instalará Pycharm que lo podemos encontrar dentro del software de Ubuntu. Vemos que es un software de licencia libre

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

**Un módulo es un archivo que contiene definiciones y declaraciones de Python**. Para la realización del mismo necesitaremos 4 archivos.

* **\_\_init\_\_.py**. Necesario para que la carpeta se trate como un paquete en Python, contiene los import de cada archivo del módulo que contenga código Python, en este caso, el archivo agenda.

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

* **­\_\_manifest\_\_.py.** Contiene un diccionario Python con la descripción del módulo, como quién es el autor, la versión del módulo o cuáles son los otros módulos de los que depende.

Texto

Descripción generada automáticamente

- name: nombre del módulo

- author: autor del módulo

- description: breve descripción de qué va a hacer el módulo

- version: por defecto 1.0

- data: fichero xml donde indicamos cómo se estructuran los datos del módulo

- depends: módulos de los que depende, en este caso al ser tan básico no depende de ninguno, pero pondremos por defecto el módulo base.

* **nombre\_modulo.py.** En este archivo creamos la clase definiendo los campos que va a tener. Al crear la clase estamos creando el modelo (una tabla en la base de datos) y también estamos creando el controlador, porque cuando creamos una clase estamos definiendo el comportamiento que tiene.

Texto

Descripción generada automáticamente

* **nombre\_modulo\_view.xml.** En este archivo definimos la vista de nuestro módulo, o del objeto que va a crear nuestro módulo. Para crear este archivo necesitamos tener ciertos conocimientos de XML, o bien coger una vista de otro objeto y a partir de ella crear la del nuevo objeto. Que es lo que yo he realizado.

Texto

Descripción generada automáticamente

Texto

Descripción generada automáticamente

Así es como queda nuestra carpeta agenda.

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

**Agregar el módulo a la ruta de complementos**

Ahora que tenemos un módulo nuevo, incluso si es muy simple, queremos que esté disponible en Odoo. Para esto, debemos asegurarnos que el directorio que contiene el módulo sea parte de la ruta de complementos **addons**. Y luego tenemos que actualizar la lista de módulos de Odoo.

Lo primero que tenemos que hacer es comprobar si en la configuración está indicada la ruta de instalación de los módulos. Para comprobarlo tenemos que ir a la carpeta donde se instaló odoo, en mi caso, creé un usuario odoo donde instale odoo, entonces para poder acceder a la carpeta iniciare mi usuario odoo y comprobaré la carpeta.

Texto

Descripción generada automáticamente

Comprobamos dentro del archivo de odoo.conf en que carpeta debemos meter nuestro modulo creado.

Texto

Descripción generada automáticamente

Ahora basta con copiar y pegar la carpeta agenda que hemos creado, se puede hacer de dos formas desde la terminal o simplemente arrastrando el archivo desde el escritorio a la carpeta addons en odoo.

Imagen que contiene PowerPoint

Descripción generada automáticamente

Observamos como se ha movido perfectamente.

Imagen que contiene Icono

Descripción generada automáticamente

Iniciamos odoo en modo desarrollador, y actualizamos la lista en las aplicaciones, escribimos agenda y vemos como aparece perfectamente.

Interfaz de usuario gráfica, Texto, Aplicación, Teams

Descripción generada automáticamente

Le damos a instalar.

Interfaz de usuario gráfica, Aplicación, Teams

Descripción generada automáticamente

Una vez instalado nos aparecerá en el menú.

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

Por defecto aparece la vista de árbol. Creamos un nuevo contacto.

Imagen que contiene Texto

Descripción generada automáticamente

En el formulario los campos requeridos aparecerán en morado como puse en el archivo agenda.py, los campos requeridos eran true, entonces aparecerán en morado de lo contrario, si pones false aparecerán en blanco.

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

Una vez introducidos los guardamos.

Imagen que contiene Interfaz de usuario gráfica

Descripción generada automáticamente

Ahora comprobamos en pgAdmin en la base de datos que se hayan introducido correctamente.

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

Hacemos una consulta seleccionamos los campos nombre y teléfono, y vemos como se han introducido perfectamente.

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente