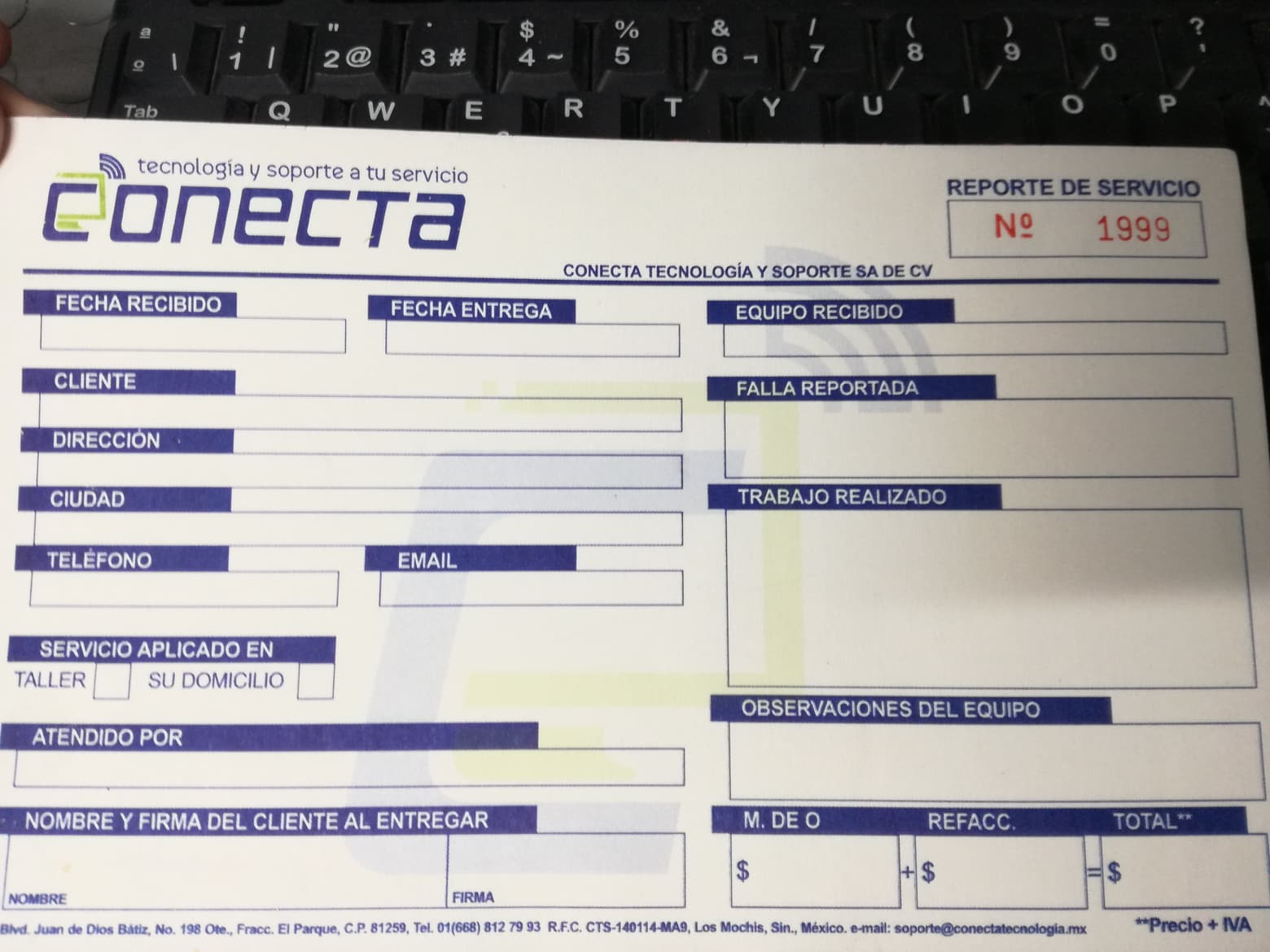
**Reporte**

Al inicio de la primera semana lo primero que realice fue hablar con el gerente el cual me presento con el resto de empleados, una vez terminada la presentación se me fueron asignadas diferentes actividades a cumplir a lo largo de mi estancia en la empresa. una de las primeras actividades fue hacer encuestas informales a los principales trabajadores que usaran el sistema en la cual pregunte los principales problemas que tiene la empresa al momento de realizar su trabajo, así como algunas propuestas que tengan ellos sobre el contenido de la aplicación con el fin de conocer como les podría ser más útil.

Después de hacer el análisis de las necesidades de la empresa y ver la forma de trabajo que manejan la mejor opción es desarrollar el sistema de forma que los empleados puedan acceder a ella desde su teléfono móvil debido a que la mayoría de los empleados salen constantemente a lo largo del día a diferentes empresas a las cuales les ofrecen sus servicios. Tomando los datos anteriores y pensando en que pueda ser útil tanto en pc como para dispositivos móviles se optó por hacer un sistema web el cual sea responsivo para que se pueda acceder a el tanto de pc como desde dispositivos móviles.

Una vez que se tiene en cuenta cuales son las necesidades de la empresa que se puedan automatizar fue necesario saber cómo es que realizan los empleados los procesos que se piensan agregar al sistema por eso fue necesario conversar sobre dichos procesos los cuales actualmente son hechos en reportes físicos como se muestra en la siguiente imagen.



Los reportes se realizan normalmente una vez terminado el trabajo, uno de los mayores riesgos con esto es la perdida de este documento físico o en algunos casos que al ir a realizar el trabajo el empleado pierda o se le olvide dichas hojas, esto ocasiona posponer los reportes de servicio lo cual hace que se pueda dar el caso de olvidar haber hecho el servicio que a su vez ocasiona una pérdida de dinero para la empresa.

Lo siguiente fue la realización de diagramas UML que en este caso fueron diagramas de casos de uso los cuales ayudaron a pensar cómo será que el sistema va a interactuar con el usuario al momento de que este haga clic en alguna opción del sistema pensando en los diferentes procesos que realiza el sistema y poder tener ya visto cuales son las todas las posibles acciones que puede hacer el usuario y saber perfectamente cómo va a actuar el sistema a todas esas acciones diferentes.

**Casos de uso**

|  |  |
| --- | --- |
| **Nombre:** | **Seleccionar opción del menú** |
| **Actor:** | El usuario |
| **Precondiciones** | Haber entrado al sistema |
| **Descripción:** | 1. El usuario entra en el menú principal. 2. El sistema muestra las opciones disponibles. 3. El usuario selecciona una de las opciones.    1. El usuario selecciona la opción registro de quipos de cómputo.    2. El usuario selecciona la opción reportar fallas. 4. El sistema manda al usuario a la pantalla seleccionada. |

|  |  |
| --- | --- |
| **Nombre:** | **Reporte de fallas** |
| **Actor:** | El usuario |
| **Precondiciones** | Seleccionar opción |
| **Descripción:** | 1. El usuario selecciona una de las opciones.    1. El usuario selecciona la opción nuevo reporte.    2. El usuario selecciona la opción buscar reporte. 2. El sistema lo envía a la ventana seleccionada.    1. El sistema manda al usuario a la pantalla de nuevo reporte.    2. El sistema manda al usuario a la pantalla de buscar reporte. 3. El usuario llena los campos solicitados.    1. El usuario llena los campos para un nuevo reporte.    2. El usuario busca un reporte mediante un id. 4. El sistema registra el nuevo reporte en la base de datos.    1. El sistema muestra el reporte requerido en la pantalla. 5. El usuario selecciona la opción salir 6. El sistema manda al usuario de vuelta a la ventana inicio. |

|  |  |
| --- | --- |
| **Nombre:** | **Registro de equipos de computo** |
| **Actor:** | El usuario |
| **Precondiciones** | Seleccionar opción |
| **Descripción:** | 1. El usuario selecciona una de las opciones disponibles.    1. El usuario selecciona registrar equipo.    2. El usuario selecciona buscar equipo. 2. El sistema lo manda a la opción seleccionada.    1. El sistema manda al usuario a la pantalla registrar equipo.    2. El sistema manda al usuario a la pantalla buscar equipo. 3. El usuario llena los campos solicitados.    1. El usuario llena los campos para registrar un nuevo equipo.    2. El usuario busca un equipo de cómputo mediante un id. 4. El sistema guarda los datos en la base de datos.    1. El sistema muestra los datos relacionados con el id de búsqueda. 5. El usuario selecciona la opción registrar otro equipo.    1. El usuario selecciona la opción salir. 6. El sistema manda al usuario a la pantalla de inicio. |
|  |  |

|  |  |
| --- | --- |
| **Nombre:** | **Cancelar registro.** |
| **Actor:** | El usuario |
| **Precondiciones** | Seleccionar opción |
| **Descripción:** | 1. El usuario selecciona la opción de cancelar registro. 2. El sistema muestra una ventana emergente con dos opciones y una advertencia de si desea abandonar el registro. 3. El usuario selecciona una de las dos opciones.    1. El usuario selecciona salir del registro.    2. El usuario selecciona continuar llenando registro. 4. El sistema manda al usuario a la ventana de inicio.    1. El usuario permanece en la ventana de registro. |

|  |  |
| --- | --- |
| **Nombre:** | **Editar registro.** |
| **Actor:** | El usuario |
| **Precondiciones** | Seleccionar opción |
| **Descripción:** | 1. El usuario selecciona la opción editar registros de acuerdo al Id del reporte. 2. El sistema lo manda a la ventana de editar registro. 3. El sistema muestra los datos del reporte seleccionado. 4. El usuario edita los cambios requeridos. 5. El usuario da clic en el botón guardar cambios. 6. El sistema guarda los datos en la base de datos. 7. El sistema manda al usuario a la ventana con la lista de los reportes existentes. |

|  |  |
| --- | --- |
| **Nombre:** | **Iniciar sesión** |
| **Actor:** | El usuario |
| **Precondiciones** | Estar dentro del sistema |
| **Descripción:** | 1. El usuario selecciona la opción iniciar sesión en cualquier pantalla. 2. El sistema manda al usuario a la pantalla de inicio de sesión. 3. El usuario escribe su cuenta y contraseña en los campos respectivos. 4. El usuario selecciona la opción iniciar sesión. 5. El sistema manda al usuario de vuelta a la pantalla en la que estaba antes de iniciar sesión. |

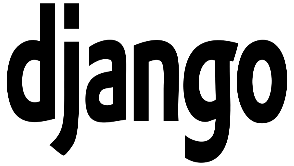
También se realizaron diagramas de interacción entre pantallas lo cual será de utilidad para conocer cómo van a interactuar las pantallas una con otras y también para tener un numero inicial de cuantas pantallas diferentes tendrá el sistema y si será necesario quitar o agregar más e incluso ver de forma más clara el contenido para saber si se pueden unir algunas pantallas y hacer más eficiente el sistema.

Una vez realizados los diagramas el siguiente paso fue elegir que se utilizaría para desarrollar el sistema para ello se realizó una investigación para poder elegir los que se puedan ser beneficiosos para el desarrollo del sistema.

Las tecnologías que se decidieron utilizar fueron las siguientes:

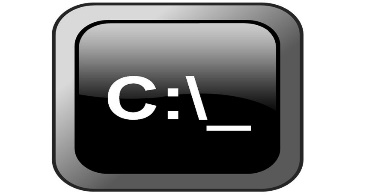
 Python es un lenguaje de programación interpretado cuya filosofía hace hincapié en una sintaxis que favorezca un código legible.

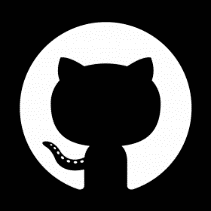
Se trata de un lenguaje de programación multiparadigma, ya que soporta orientación a objetos, programación imperativa y, en menor medida, programación funcional. Es un lenguaje interpretado, usa tipado dinámico y es multiplataforma.

 Django es un framework de desarrollo web de código abierto, escrito en Python, que respeta el patrón de diseño conocido como Modelo–vista–template. La meta fundamental de Django es facilitar la creación de sitios web complejos. Django pone énfasis en el re-uso, la conectividad y extensibilidad de componentes, el desarrollo rápido y el principio No te repitas (DRY, del inglés Don't Repeat Yourself). Python es usado en todas las partes del framework, incluso en configuraciones, archivos, y en los modelos de datos.

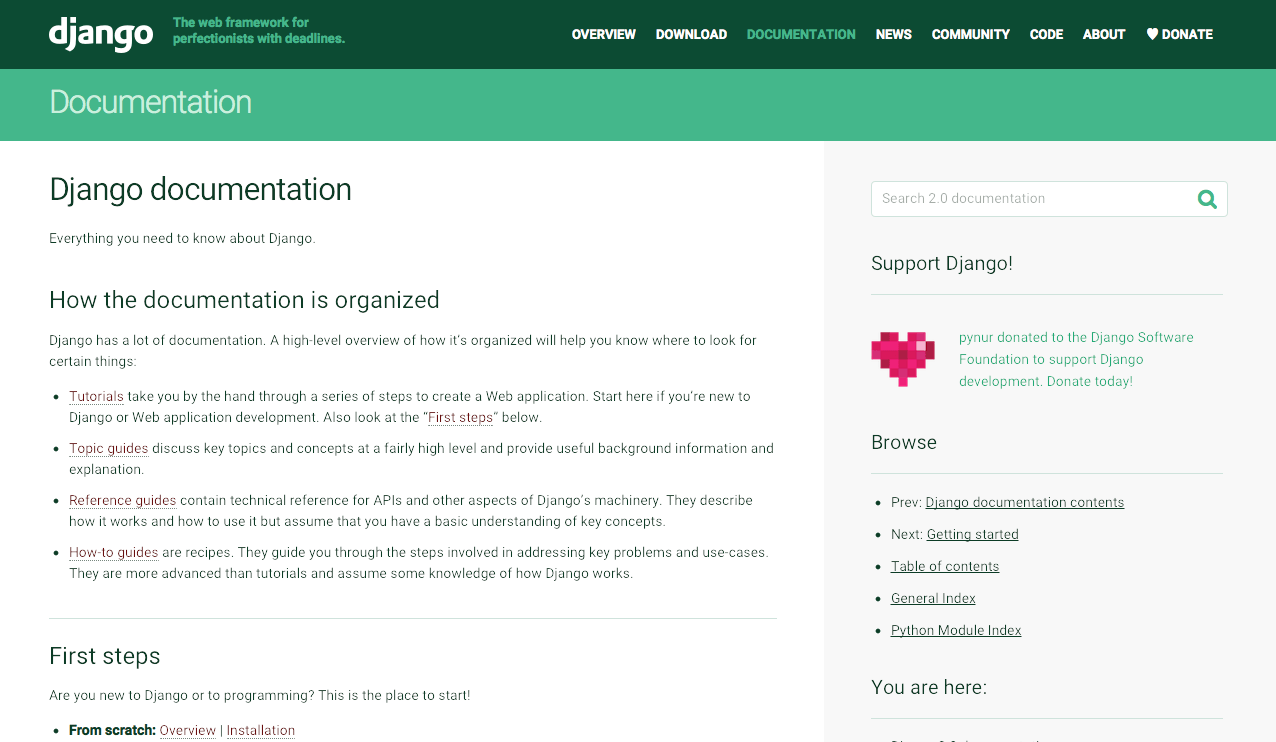
 MySQL es un sistema de gestión de bases de datos relacional desarrollado bajo licencia dual: Licencia pública general/Licencia comercial por Oracle Corporation y está considerada como la base datos de código abierto más popular del mundo.

 Sublime Text es un editor de texto y editor de código fuente está escrito en C++ y Python para los plugins.1​ Desarrollado originalmente como una extensión de Vim, con el tiempo fue creando una identidad propia, por esto aún conserva un modo de edición tipo vi llamado Vintage mode.

 El símbolo del sistema (en inglés, Command prompt, también conocido como cmd.exe o simplemente cmd) es el intérprete de comandos en OS/2 y sistemas basados en Windows NT (incluyendo Windows 2000, Windows XP, Windows Server 2003, Windows Vista, Windows 7, Windows 8, Windows 8.1 y Windows 10). Es el equivalente de COMMAND.COM en MS-DOS y sistemas de la familia Windows 9x.

 GitHub es una forja (plataforma de desarrollo colaborativo) para alojar proyectos utilizando el sistema de control de versiones Git. Se utiliza principalmente para la creación de código fuente de programas de computadora.

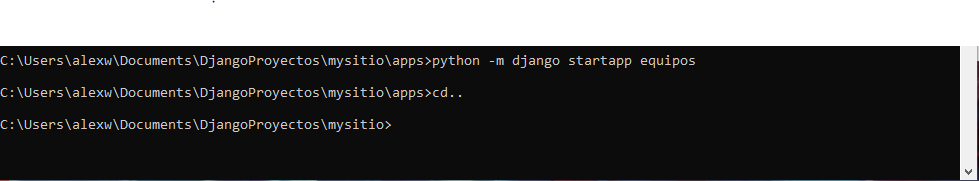
Una vez teniendo las tecnologías que se utilizaran el siguiente paso fue instalarlas en el equipo de cómputo que se utilizara para desarrollar el sistema. Después de las instalaciones se realizó una pequeña capacitación sobre python y Django realizando lectura de su documentación y haciendo pequeñas pruebas de su funcionamiento.



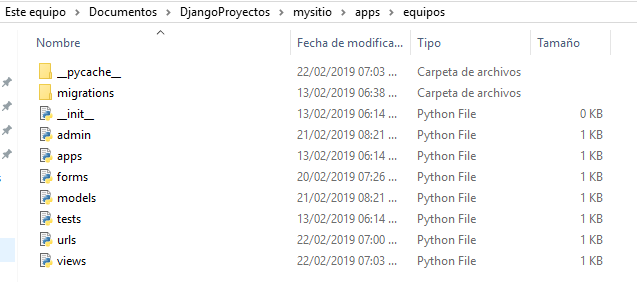
Al empezar con la elaboración del sistema fue la creación de la carpeta donde se guardara el proyecto y posteriormente sincronizarla con Github.



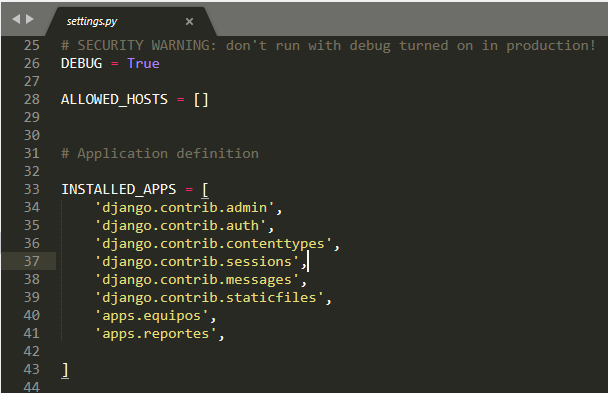
Después fue necesario crear las carpetas de donde se escribirá el código de las diferentes acciones que hará el sistema para crear estas carpetas es necesario crearlas desde símbolos del sistema ya que será necesario que se creen con archivos con la extensión de Python.



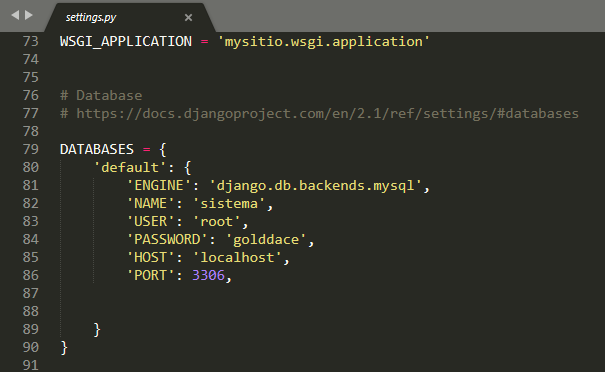
Después de crearlo se genera una carpeta en la ruta establecida con los siguientes archivos:



Una vez creadas es necesario agregar la ruta de las carpetas, denominadas en Django Apps, al archivo Setings.py del sistema para que puedan ser utilizadas al momento de correr el servidor.

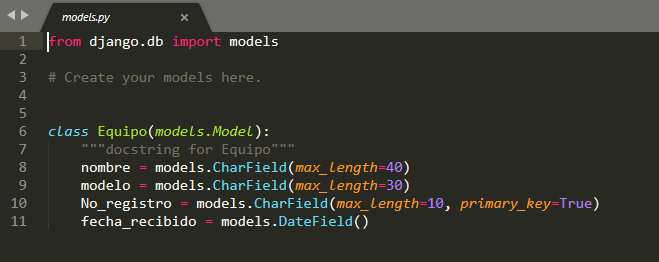
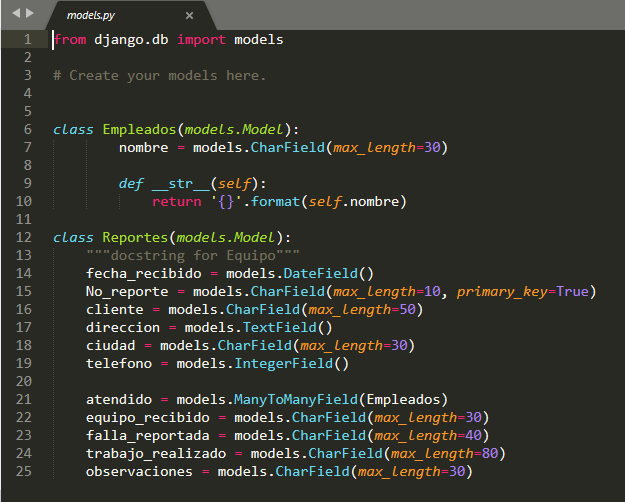


El siguiente paso fue crear la base de datos del sistema y realizar la conexión con el sistema web para lo cual es necesario volver a entrar al archivo Setings,py del sistema para realizar la conexión llenando los campos con la información de la base de datos.

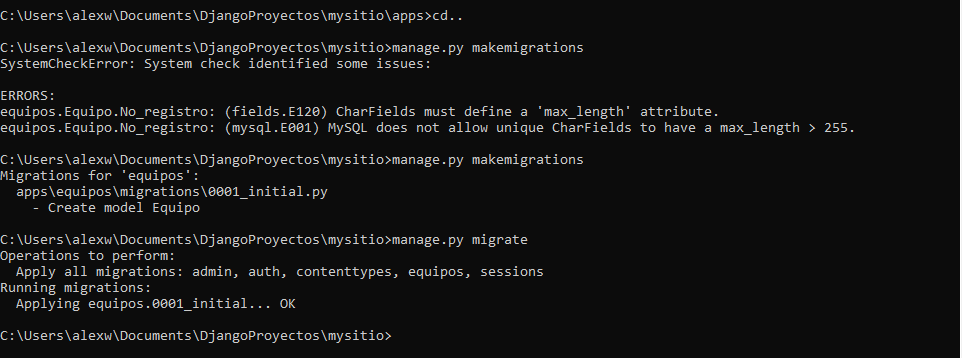


Como por el momento todas las pruebas del sistema son locales en la parte de “HOST” tenemos asignado ‘localhost’ lo cual será cambiado una vez que se tenga el servidor para el sistema.

Después de realizar la conexión con la base de datos lo siguiente fue crear los modelos que son utilizados para crear las tablas de la base de datos los modelos son creados en los archivos de models.py correspondiente a cada app.

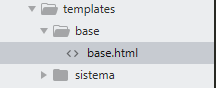


Una vez hechas las clases con los campos que tendrán las tablas desde CMD(Símbolos de sistema) se ejecutan los siguientes comandos.



Los cuales lo que hacen es hacer una migración de los modelos creados a la base de datos a la cual está conectada. Después de comprobar que las tablas se hayan creado con éxito el siguiente paso fue crear una carpeta nueva llamada templates la cual contendrá todos los archivos con la extensión .html del sitio que son donde se visualizara el sistema desde cualquier navegador.

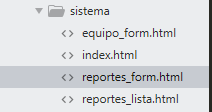
La carpeta base contiene una plantilla base en la cual se usara para todas las páginas que se utilizaran en el sistema.



Para hacer referencia a base.html en otro archivo se escribe la siguiente etiqueta.



En la carpeta de sistema se crearan todas las páginas que se utilizaran en el sistema de forma que se irán agregando conforme sean necesarias.

 **IMG10**

Para poder utilizar las vistas de las apps es necesario agregar las rutas a url.py de su app correspondiente. **IMG11**



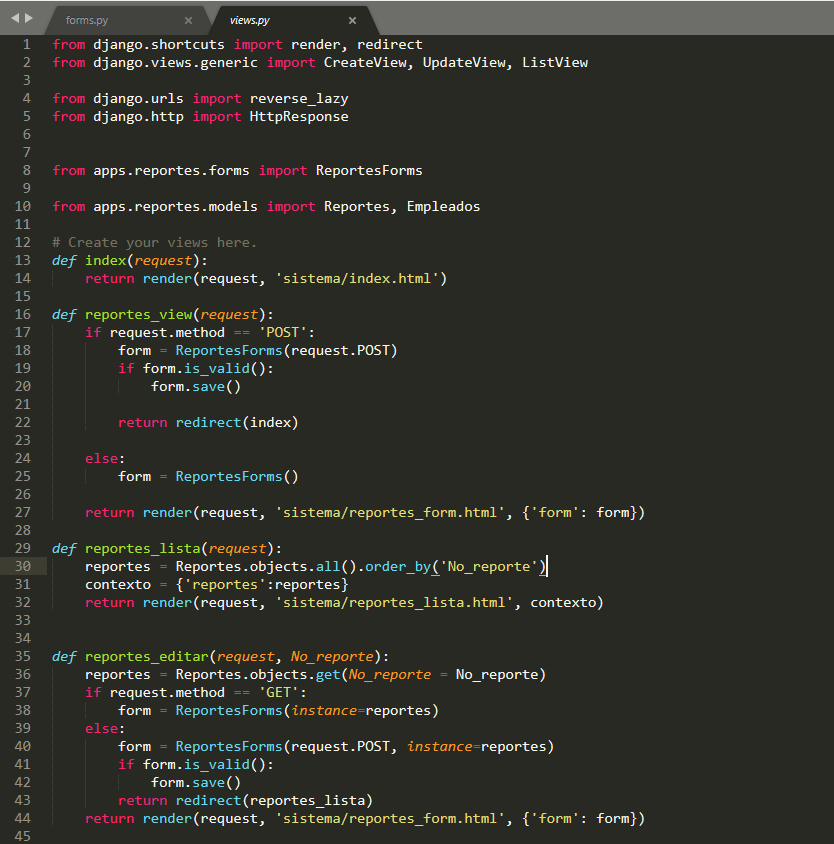
Ahora para comenzar a desarrollar los formularios del sistema es necesario situarse en forms.py de la app en la que se realizara el formulario en este caso en la carpeta de reportes.

Se crea una clase en la cual la variable model hace referencia al modelo que hace referencia a los campos que tendrá el formulario.

**Fields.** Son los textos que se mostraran en el formulario.

**Labels.** En esta parte se le asigna a cada field el nombre del campo que va a llenar de los modelos en la base de datos.

**Widgets.** Leasigna a cada field el campo de entrada de texto que tendrá.

Después de terminar las clase en forms.py nos dirigimos a views.py en la cual se definen los métodos que serán utilizados.

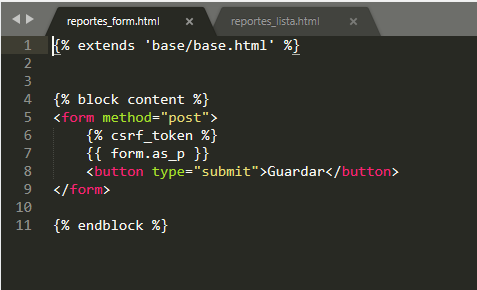
**Def index.** Lo que haces es que cuando se acceda a una de las rutas previamente asignadas en IMG11 te mande a la página indicada en este caso mandara a index.html.

**Def reportes\_view.** Hace un método en el cual lo que hará será guardar los datos de los campos llenados en este caso en la página de reportes/nuevo (IMG14) después verifica si todos los datos son válidos para posteriormente guardarlos en la base de datos. En caso de que los datos sean correctos te mandara a la página principal de lo contrario mandara un mensaje de que dato está mal y en qué campo.

**Def reportes\_listar.** Hace una lista de todos los reportes guardados en la base de datos.(IMG15)

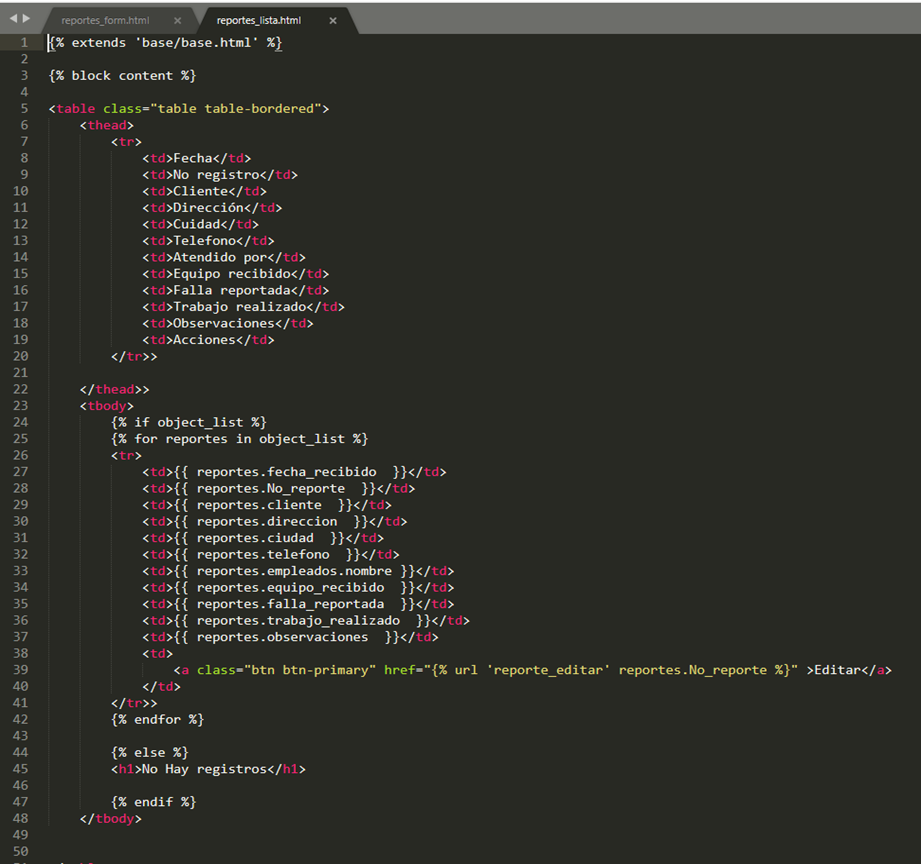
**Def reportes\_editar.** Hace edición sobre algún reporte ya existente mediante el método GET el cual toma el valor de la llave primaria para tomar los demás datos de un reporte y mostrarlo al usuario.

Una vez definidos los métodos lo siguientes es situarnos en los archivos .HTML para agregar los métodos que se utilizaran en este caso en **reportes\_forms**.html será para crear nuevos reportes con el método “POST” que fue definido en **reportes\_view**.

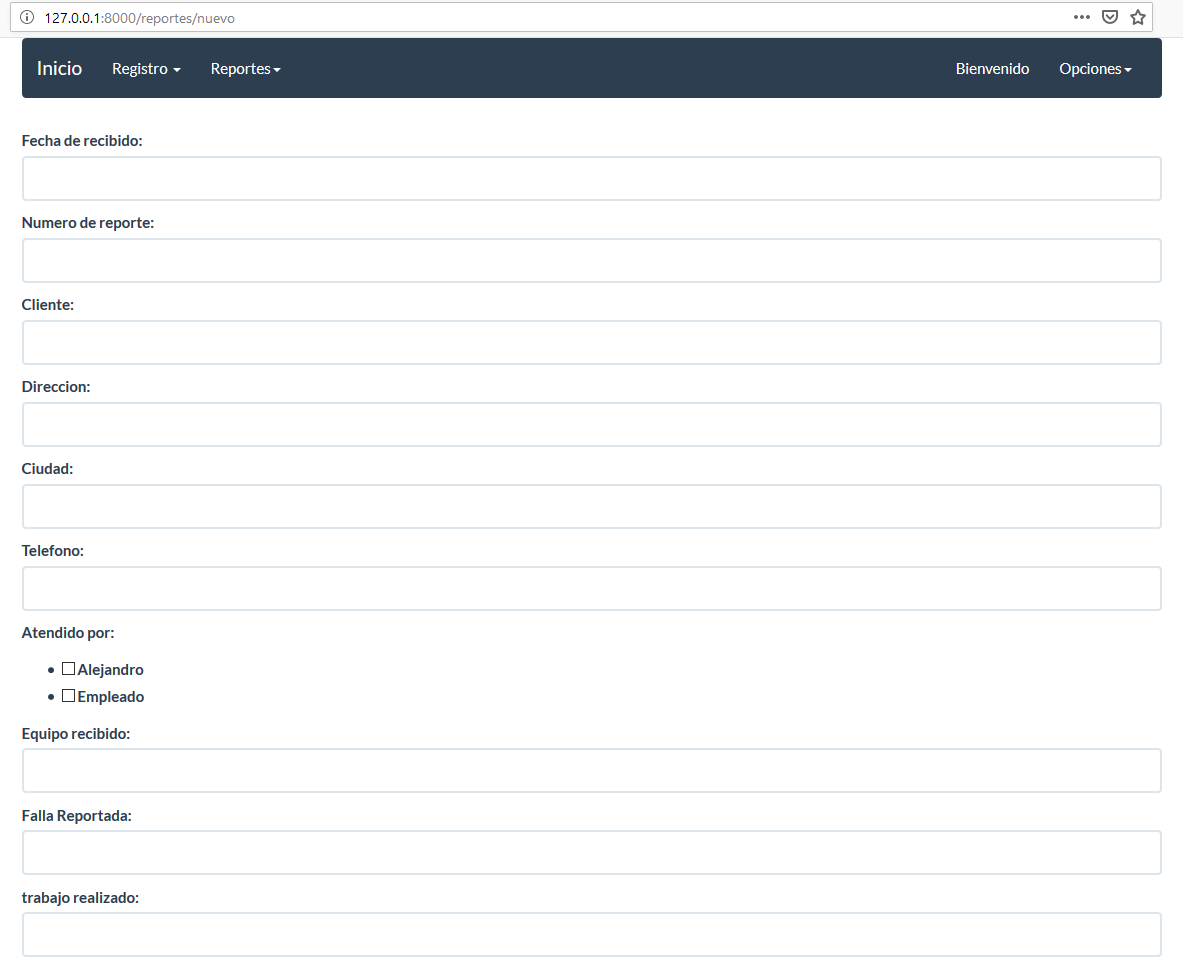
****

**csrf\_token** etiqueta dentro del <form>elemento si el formulario es para una URL interna. Esta etiqueta es necesaria de lo contrario no se podrá acceder al formulario.

En cuanto a la página que contiene la lista de los reportes se creara una tabla con los reportes que están guardados en la base de datos en la tabla correspondiente en este caso de la tabla de reportes.

****

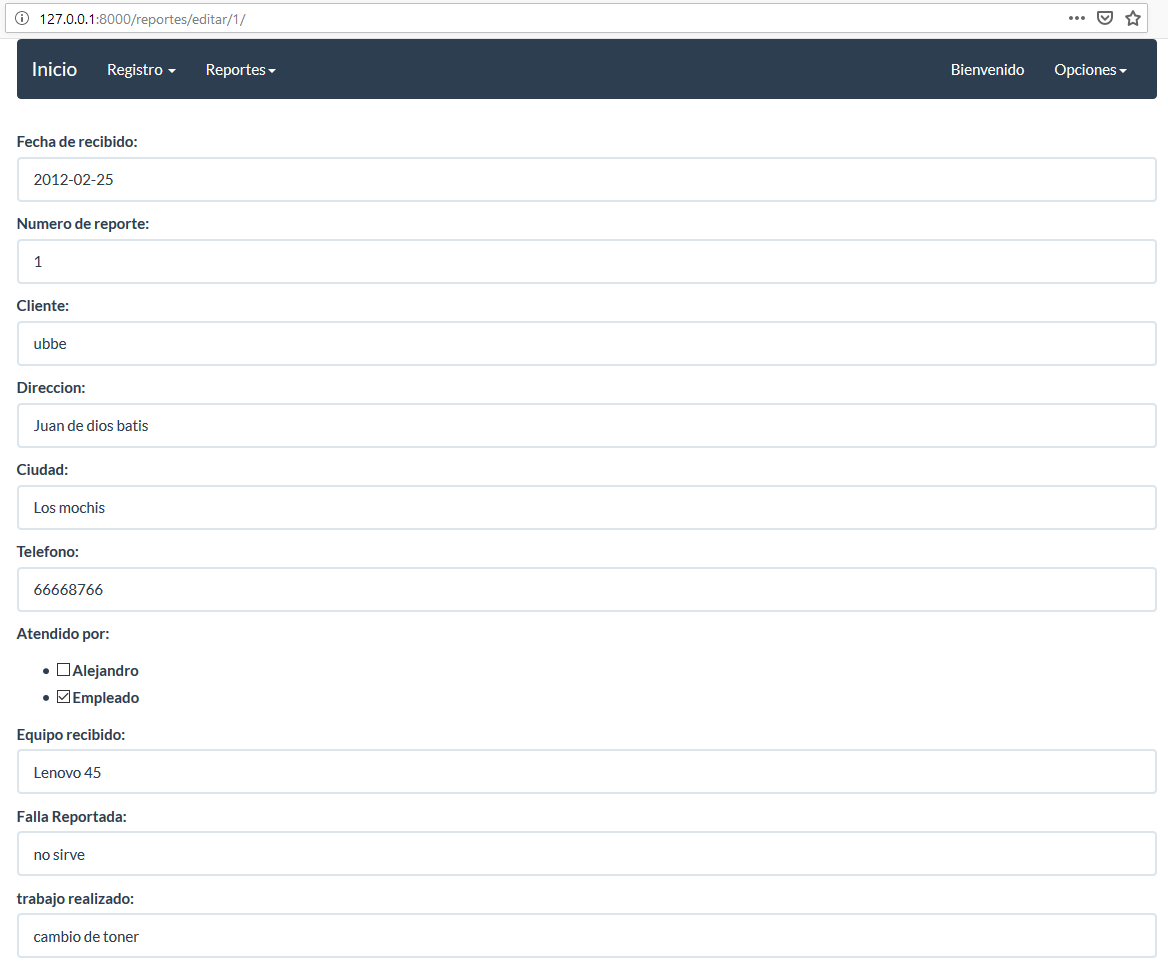
**Vista de la página para llenar un reporte nuevo. (IMG14)**

****

**Vista de la página con la lista de los reportes guardados. (IMG15)**

****

**Para acceder a editar un reporte se hace desde la página de lista la cual contiene un botón el cual permite editar un reporte en específico.**

****