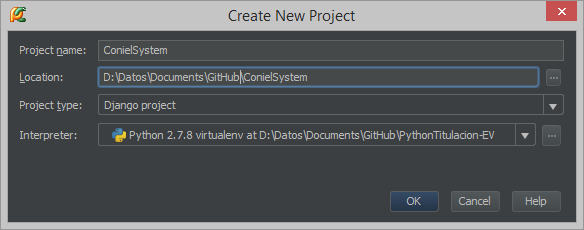
ESTRUCTURA E IMPLEMENTACION DE LA APLICACIÓN

La estructura utilizada para el desarrollo de la aplicación con el framework Django en su versión 1.6 bajo el Ide de desarrollo Pycharm se muestra a continuación en la figura…

En la figura … se muestra la organización del proyecto utilizando el esquema que nos provee el framework para el desarrollo adaptándolo para la integración al sistema Personal Communications Terminal de IBM, además nos muestra que cada parte del proyecto tiene su directorio en específico para promover un código limpio y estable.

EJEMPLO DE IMPLEMENTACIÓN DE UN CASO DE USO

A continuación se muestra la implementación del caso de Uso de Login especificado en la primera fase de la metodología.

* Se crea el proyecto de django mediante el id de desarrollo Pycharm.
* Se modifica las configuraciones del proyecto contenidas en el fichero settings.py para la ejecucion correcta del servidor en base los puntos predefinidos que seguirá la aplicación.

...

# Database

# https://docs.djangoproject.com/en/1.6/ref/settings/#databases

DATABASES = {

'default': {

'ENGINE': 'django.db.backends.postgresql\_psycopg2',

'NAME': 'Coniel',

'USER': 'postgres',

'PASSWORD': '\*\*\*\*\*\*\*',

'HOST': '127.0.0.1',

'PORT': '5432',

},

}

# Internationalization

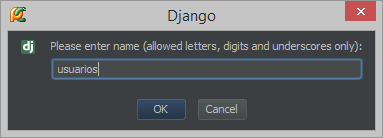
# https://docs.djangoproject.com/en/1.6/topics/i18n/

LANGUAGE\_CODE = 'es-ec'

TIME\_ZONE = 'America/Guayaquil'

...

* Se crea la app de la misma manera, ésta se encarga de gestionar los usuarios y los accesos a la aplicación.



* Se registra la app recientemente creada en nuestro proyecto

Settings.py

...

# Application definition

INSTALLED\_APPS = (

'django.contrib.auth',

'django.contrib.contenttypes',

'django.contrib.sessions',

'django.contrib.messages',

'django.contrib.staticfiles',

'suit',

'django.contrib.admin',

'django\_ajax',

'bootstrap3',

'serviciosWeb',

'busquedas',

'usuarios',

'ingresos',

'inventario',

'django\_extensions',

)

...

* Tomando como base el proyecto de django se modifican los ficheros que se nos proporcionan para el derarrollo, se comienza a aplicar la lógica mediante modelos, los mismos que son la definición de las clases obtenidas en las fases previas de la presente metodología.

Usuarios/models.py

class usuarioSico(models.Model):

nombre = models.CharField(max\_length=10, verbose\_name='Nombre de Usuario en Sico')

clave = models.CharField(max\_length=10, verbose\_name='Contraseña en Sico')

contrato = models.ForeignKey('inventario.contrato')

def \_\_unicode\_\_(self):

return '%s - %s' % (self.nombre, self.contrato.num)

class Meta:

verbose\_name='Usuario en Sico'

verbose\_name\_plural='Usuarios en Sico'

#MODICANDO EL MODELO POR DEFECTO USER

User.add\_to\_class('sesion\_sico', models.CharField(max\_length=2, null=True, blank=True))

User.add\_to\_class('usuario\_sico', models.ManyToManyField(usuarioSico, blank=True, null=True))

Inventario/models.py

class contrato(models.Model):

num = models.CharField(max\_length=10, primary\_key=True, verbose\_name='Número de Contrato')

descripcion = models.CharField(max\_length=150, verbose\_name='Descripción')

zonas = models.CharField(max\_length=150, verbose\_name='Zona(s)')

codigoInstalador = models.PositiveSmallIntegerField(verbose\_name='Código de Instalador')

inicioVigencia = models.DateField(verbose\_name='Vigente Desde')

finalVigencia = models.DateField(verbose\_name='Vigente Hasta')

def \_\_unicode\_\_(self):

return self.num

* A continuación se registra las urls que son identificadas como válidas para esta aplicación editando el fichero urls.py.

admin.autodiscover()

urlpatterns = patterns('',

#Administración

url(r'^admin/', include(admin.site.urls)),

#Usuarios

url(r'^$', 'usuarios.views.main', name='main'),

url(r'^login$', "usuarios.views.ingreso", name='login'),

url(r'^logout$', 'usuarios.views.salir', name="logout"),

url(r'^home$', 'usuarios.views.home', name='home'),

* A continuación se modifica las vistas especificadas para cada url de esta manera se gestiona las peticiones de los usuarios.

def main(request):

#return render\_to\_response('main.html', {}, context\_instance=RequestContext(request))

return ingreso(request)

@login\_required()

def salir(request):

cerrarSico(request.user.sesion\_sico)

request.user.sesion\_sico = ''

request.user.save()

logout(request)

return HttpResponseRedirect('/login')

def cerrarSico(sesionAct):

c = manejadorDeConexion()

c.closeProgram(sesionAct)

def ingreso(request):

error = None

try:

if request.user.sesion\_sico:

return home(request)

else:

salir(request)

except:

pass

if request.POST:

form = LogIn(request.POST)

if form.is\_valid():

username = form.cleaned\_data["usuario"]

password = form.cleaned\_data["clave"]

contrato = form.cleaned\_data["sico"]

user = authenticate(username=username, password=password)

if user is not None:

if user.is\_active:

if user.sesion\_sico:

error = "El Usuario especificado ya esta en uso."

try:

logout(request)

except:

pass

else:

try:

u = usuarioSico.objects.get(user=user, contrato=contrato)

except:

u = False

if isinstance(u, usuarioSico):

if integracion(u.nombre, u.clave, user):

login(request, user)

request.session['contrato'] = contrato

return HttpResponseRedirect('/home')

else:

error = 'El Sistema Comercial(Sico Cnel) no esta disponible por el momento...'

user.sesion\_sico=''

user.save()

else:

error = 'El Usuario Especificado no cuenta con permisos necesarios para acceder al contarto'

elif username and password:

error = "Su Usuario o Contraseña no son correctos, Intentelo nuevamente."

form = LogIn()

return render\_to\_response('usuarios/login.html', {'f': form, 'errors': error},

context\_instance=RequestContext(request))

@login\_required()

def home(request):

print 'Sesion : %s' % request.user.sesion\_sico

if request.user.sesion\_sico:

return render\_to\_response('usuarios/home.html', {}, context\_instance=RequestContext(request))

return HttpResponseRedirect('/login')

def integracion(u, c, user):

conn = None

while True:

conn = manejadorDeConexion()

user.sesion\_sico = str(conn.getAvailableConnection())

if not User.objects.filter(sesion\_sico=user.sesion\_sico):

user.save()

break

return conn.openSession(connectionName=user.sesion\_sico, usuario=u, contrasenia=c)

* Adicionalmente a través de la siguiente clase se gestionan las peticiones al terminal de PCOMM, y de esta manera logramos que nuestro servidor abra una terminal de datos por cada usuario logeado en el sistema.

import pythoncom

\_\_author\_\_ = 'Jhonsson'

from time import sleep

import win32com

import win32com.client

class manejadorDeConexion:

def \_\_init\_\_(self):

pythoncom.CoInitialize()

self.PCommConnMgr = win32com.client.Dispatch('PCOMM.autECLConnMgr')

self.connList = self.PCommConnMgr.autECLConnList

self.activeSession = None

self.activeConnection = ''

self.estado = False

def setActiveSession(self, connectionName):

self.activeSession = win32com.client.Dispatch("PCOMM.autECLSession")

if connectionName:

self.activeConnection = connectionName

self.activeSession.SetConnectionByName(self.activeConnection)

# return value from screen at position (row, col)

def getText(self, row, col, length=None, connectionName=None):

if connectionName is not None:

self.setActiveSession(connectionName)

if length is None:

self.activeSession.autECLPS.autECLFieldList.Refresh()

field = self.activeSession.autECLPS.autECLFieldList.FindFieldByRowCol(row, col)

length = field.Length

result = self.activeSession.autECLPS.GetText(row, col, length)

return result

def sendKeys(self, count, key, row=None, col=None, connectionName=None, wait=True):

n = 0

if connectionName is not None:

self.setActiveSession(connectionName)

while n < count:

if row is None or col is None:

self.activeSession.autECLPS.SendKeys("%s" % key)

else:

self.activeSession.autECLPS.SendKeys("%s" % key, row, col)

if wait:

if not self.activeSession.autECLOIA.WaitForAppAvailable(5000): return False

if not self.activeSession.autECLOIA.WaitForInputReady(5000): return False

n += 1

return True

def getAvailableConnection(self):

self.connList.Refresh()

self.activeConnection = chr(self.connList.Count + 65)

self.openProgram()

self.connList.Refresh()

return self.activeConnection

def openSession(self, connectionName=None, usuario="none", contrasenia="none"):

self.estado = False

if connectionName is not None:

self.setActiveSession(connectionName)

else:

self.setActiveSession(self.getAvailableConnection())

segundos = 10

while not self.activeSession.Ready and segundos >= 0:

print "la session no esta lista aun ..."

#return False

sleep(1)

segundos -= 1

if segundos >= 0:

if self.activeSession.autECLOIA.WaitForAppAvailable(5000):

if self.activeSession.autECLOIA.WaitForInputReady(5000):

print self.getText(22, 50, length=8)

print self.getText(21, 50, length=12)

if self.activeSession.autECLPS.WaitForString('USUARIO', 21, 50, 10000):

if self.sendKeys(1, usuario, row=21, col=63):

if self.sendKeys(1, contrasenia, row=22, col=63):

if self.sendKeys(6, '[enter]'):

self.estado = True

return self.estado

self.closeProgram(self.activeConnection, directo=True)

return self.estado

def openProgram(self):

print "Abriendo el Programa, Sesion: {0}".format(self.activeConnection)

self.PCommConnMgr.StartConnection(

"PROFILE=.\ControlSystem\pComm\sico\CNEL.WS CONNNAME={0} WINSTATE=MIN".format(self.activeConnection))

def closeProgram(self, connection, directo=False):

if not connection:

connection = self.activeConnection

try:

print "Cerrando el Programa, Sesion: {0}".format(connection)

if not directo:

if not self.activeSession:

self.setActiveSession(connection)

self.sendKeys(10, '[pf12]', connectionName=connection, wait=False)

self.PCommConnMgr.StopConnection(connection, "saveprofile=no")

except:

print "No se ha podido cerrar la sesion: {0}".format(connection)

* Se escriben los ficheros html que son interpretados por los navegadores de la siguiente manera dentro del directorio de templates del proyecto

templates/base.html

{% load staticfiles %}

{% load bootstrap3 %}

<!DOCTYPE html>

<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="es" lang="es-ec">

<head>

<meta charset="utf-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1">

<title id="title">

{% block title %}

P&#225;gina Sin T&#237;tulo

{% endblock %}

</title>

<!-- CSS de Bootstrap -->

<link href="{% static 'css/bootstrap.min.css' %}" rel="stylesheet" media="screen">

<link href="{% static 'css/bootstrap-theme.min.css' %}" rel="stylesheet" media="screen">

<!-- CSS de JQuery-UI -->

<link href="{% static 'css/jquery-ui.min.css' %}" rel="stylesheet" media="screen">

<link href="{% static 'css/jquery-ui.structure.min.css' %}" rel="stylesheet" media="screen">

<link href="{% static 'css/jquery-ui.theme.min.css' %}" rel="stylesheet" media="screen">

<!-- CSS Personal -->

<link href="{% static 'css/jhonsson.css' %}" rel="stylesheet" media="screen">

<!-- Favicon-->

<link id="favicon" rel="icon" type="image/x-icon" sizes="16x16" href="{% static 'favicon/favicon.ico' %}">

<link id="favicon2x" rel="icon" type="image/png" sizes="32x32" href="{% static 'favicon/favicon.png' %}">

<link rel="shortcut icon" type="image/png" href="{% static 'favicon/favicon.png' %}">

<!-- JS Librería jQuery -->

<script src="{% static 'js/jquery-1.11.1.js' %}"></script>

<script src="{% static 'js/jquery-ui.min.js' %}"></script>

<!-- JS de Bootstrap -->

<script src="{% static 'js/bootstrap.min.js' %}"></script>

<script src="{% static 'js/bootstrap-hover-dropdown.min.js' %}"></script>

<script src="{% static 'js/jhonsson.js' %}"></script>

<!-- JS Librería jQueryDajax respuestas asincronas -->

<script src="{% static 'js/jquery.ajax.js' %}"></script>

<script src="{% static 'js/jquery.ajax-plugin.js' %}"></script>

<script src="{% static 'js/jquery.dajax.core.js' %}"></script>

<!-- librerías opcionales que activan el soporte de HTML5 para IE8 -->

<!--[if lt IE 9]>

<script src="https://oss.maxcdn.com/libs/html5shiv/3.7.0/html5shiv.js"></script>

<script src="https://oss.maxcdn.com/libs/respond.js/1.4.2/respond.min.js"></script>

<![endif]-->

{% block jquery %}

<script>

var dx=$('document');

dx.ready(navegacion);

function navegacion()

{

$('#{{ nav }}').addClass('active');

}

</script>

{% endblock %}

</head>

<body onLoad="window.setTimeout('hora()',1000);">

<div id="wrap">

<nav class="navbar navbar-default navbar-fixed-top" role="navigation">

<div style="padding-right: 20px">

<div class="navbar-header">

<a href="{% url 'home' %}">

<img src="{% static 'img/logo2.png' %}" height="69px" width="240px">

</a>

</div>

<div class="navbar-collapse collapse">

{% block navTop %}

{% include "navTop.html" %}

{% endblock %}

</div><!--/.nav-collapse -->

</div>

</nav>

<div class="container-fluid">

<div class="row" >

<div class="col-sm-3 col-md-2 sidebar">

{% block navLeft %}

{% include "navLeft.html" %}

{% endblock %}

</div>

<div class="col-sm-9 col-md-10 main">

{% block content %}

<h1>Contenido</h1>

{% endblock %}

</div>

</div>

</div>

<div id="push"></div>

</div>

<footer id="footer" class="footer">

<div class="container text-center">

<p>Coniel System V 1.0 - 2014.<br>

<a href="#Acercade" class="text-info">Acerca de</a></p>

{% block footer %}

{% endblock %}

</div>

</footer>

{% block jqueryfooter %}

{% endblock %}

</body>

</html>

Templates/usuarios/Login.html

{% extends "base.html" %}

{% load bootstrap3 %}

{% load static %}

{% block title %}

Acceso al Sistema

{% endblock %}

{% block content %}

<div class="row">

<div class="col-sm-11 col-md-9">

<form role="form" class="form-signin" method="post" id="login-form" action="{% url 'login' %}">

{% csrf\_token %}

<div class="panel panel-primary">

<div class="panel-heading">

<h8>

<span class="glyphicon glyphicon-log-in icon-white"></span>

Inicio se Sesi&oacute;n

</h8>

</div>

<div class="panel-body">

<div class="form-row control-group">

{{ f.usuario.label }}:

<div class="controls">

{% bootstrap\_field f.usuario layout='inline' %}

</div>

</div>

<div class="form-row control-group">

{{ f.clave.label }}:

<div class="controls">

{% bootstrap\_field f.clave layout='inline' %}

</div>

</div>

<div class="form-row control-group">

{{ f.sico.label }}:

<div class="controls">

{% bootstrap\_field f.sico layout='inline' %}

</div>

</div>

<div class="form-row submit-row">

<input type="submit" class="btn btn-info" value="Ingresar"/>

</div>

{% if errors %}

<br>

<div class="alert alert-danger alert-dismissible">

<button type="button" class="close" data-dismiss="alert">&times;</button>

<label class="control-label small" for="inputError">

{{ errors }}

</label>

</div>

{% endif %}

</div>

</div>

<br>

<input type="hidden" name="next" value="{{ next }}"/>

</form>

</div>

<div class="col-sm-1 col-md-3"></div>

</div>

{% endblock %}

Templates/usuarios/home.html

{% extends 'base.html' %}

{% load staticfiles %}

{% block title %}

{{ user.first\_name }} {{ user.last\_name }}

{% endblock %}

{% block content %}

<div class="row" style="margin-top: 50px">

<div class="col-md-4 text-center">

<button type="button" class="btn btn-link btn-lg btn-block" onclick="newUrl('{% url 'buscarCuenta' %}')">

<img class="img-responsive center-block" src="{% static 'icons/Light\_Search.png' %}"><h4>Buscar en Sico</h4>

</button>

</div>

<div class="col-md-4 text-center">

<button type="button" class="btn btn-link btn-lg btn-block" onclick="newUrl('{% url 'ingresarSico' %}')">

<img class="img-responsive center-block" src="{% static 'icons/Light\_File.png' %}"><h4>Ingresar Información al Sistema</h4>

</button>

</div>

<div class="col-md-4 text-center">

<button type="button" class="btn btn-link btn-lg btn-block" onclick="newUrl('{% url 'listadeingresos' %}')">

<img class="img-responsive center-block" src="{% static 'icons/Light\_Accept.png' %}"><h4>Actividades Realizadas</h4>

</button>

</div>

</div>

<div class="row" style="margin-top: 50px">

<div class="col-md-4 text-center">

<button type="button" class="btn btn-link btn-lg btn-block" onclick="newUrl('{% url 'cuadrillas' %}')">

<img class="img-responsive center-block" src="{% static 'icons/Light\_Users.png' %}"><h4>Monitoreo de Cuadrillas</h4>

</button>

</div>

<div class="col-md-4 text-center">

<button type="button" class="btn btn-link btn-lg btn-block" onclick="newUrl('{% url 'reportes' %}')">

<img class="img-responsive center-block" src="{% static 'icons/Light\_File\_edit.png' %}"><h4>Reportes</h4>

</button>

</div>

<div class="col-md-4 text-center">

<button type="button" class="btn btn-link btn-lg btn-block">

<img class="img-responsive center-block" src="{% static 'icons/Light\_Percent.png' %}"><h4>Avances de Obra</h4>

</button>

</div>

</div>

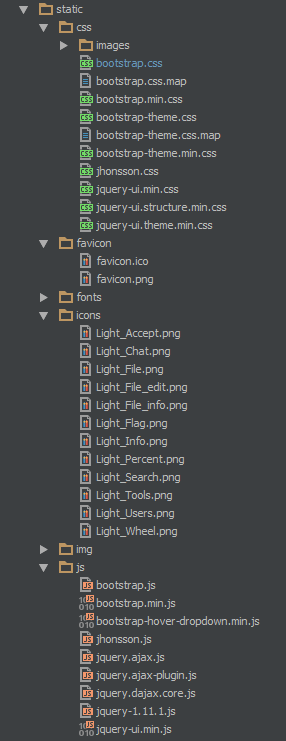
{% endblock %}

{% block footer %}

<!--footer-->

{% endblock %}

* Se utiliza el siguiente esquema para los ficheros estáticos del proyecto de entre los cuales se destaca el uso de Jquery y Bootstrap para añadir dinamismo y diseño a las paginar html.



* Finalmente se sincroniza con la base de datos los modelos descritos y se pone en marcha el servicio para comprobar el funcionamiento de la solución al primer caso de uso





