

iPresence

Adm. de Sistemas Informáticos en Red
PROYECTO INTEGRADO

Antonio Ruiz Rondán
Curso 2012-2013

Indice

1.- ESTUDIO DEL PROBLEMA Y ANÁLISIS DEL SISTEMA	3
1.1.- Introducción	3
1.2.- Objetivos.....	3
1.3.- Funciones y rendimientos deseados.....	3
1.4.- Planteamiento y evaluación de diversas soluciones	7
1.5.- Justificación de la solución elegida	8
1.6.- Modelado de la solución	9
1.6.1.- Recursos humanos.....	9
1.6.2.- Recursos hardware.....	9
1.6.3.- Recursos software.....	10
1.7.- Planificación temporal.....	11
2.- EJECUCIÓN DEL PROYECTO.....	11
2.1.- Modelo de Datos Entidad-Relación (E/R).....	12
2.1.1.- La entidad USER.....	13
2.1.2.- ALUMNO, TUTOR_DOCENTE y TUTOR_LABORAL.....	14
2.1.3.- CENTRO y EMPRESA.....	15
2.1.4.- GERENTE.....	16
2.1.5.- APRENDIZAJES.....	17
2.1.6.- CONTRATOS_PROGRAMAS.....	18
2.1.7.- SEGUIMIENTOS.....	19
2.1.8.- VISITAS.....	20
2.1.9.- USUARIOS-INACTIVOS.....	20
2.1.10.- DDL de la aplicación.....	22
2.2.- Aplicación Tele-presencia.....	25
2.2.1.- ROOMS.....	25
2.2.2.- DDL tele-visita.....	25
2.2.3.- Arquitectura de Opentok.....	26
2.2.4.- WebRTC	27
3.- FASE DE PRUEBAS.....	28
4.- DOCUMENTACION DEL SISTEMA.....	30
4.1.- Introducción a la aplicación.....	30
4.2.- Manual de instalación.	30
4.3.- Manual de administración.	42
4.4.- Manual de usuario.....	44
4.4.1.- Manual del tutor docente.....	44
4.4.2.- Manual del alumno.....	48
4.4.3.- Manual del tutor laboral.....	48
4.4.4.- Videoconferencia.....	49
5.- CONCLUSIONES FINALES.....	52
6.- BIBLIOGRAFIA.....	52

1.- ESTUDIO DEL PROBLEMA Y ANÁLISIS DEL SISTEMA

1.1.- Introducción

Mi proyecto de fin de ciclo tiene como nombre iPresence, es una aplicación web de Tele-presencia de FCT (Formación en Centros de Trabajo) para los alumnos del CFGS de Administración de Sistemas Informáticos del IES Fernando Aguilar Quignón.

1.2.- Objetivos

La idea de realizar el presente proyecto surge como consecuencia de la necesidad de disponer de una herramienta que fundamentalmente cumpla los siguientes objetivos:

- Tele-visitar al alumno y/o tutor laboral. Lo cual se se traduce en:
 - Reducción notable en el tiempo invertido en las visitas a los centros de trabajo, incrementándose por tanto la productividad de los agentes implicados
 - Ahorro por tanto en los gastos en viajes, que en tiempos de crisis no es algo a despreciar.
 - Aumenta la facilidad para reunirse con varios alumnos en la misma jornada, pudiéndose producir un intercambio de ideas más rápido y frecuente.
- Centralizar la información: Con esto se tendrá una mayor coherencia de los datos, así como una mayor manejabilidad, al estar toda esta información en una base de datos relacional.
- Generar informes derivados de la FCT: Dichos informes tendrán el mismo formato y estructura que aquellos documentos disponibles para el seguimiento de la FCT, dichos documentos están en formato PDF rellenable, disponibles para su descarga en Séneca.

1.3.- Funciones y rendimientos deseados

El resultado de este proyecto por tanto será una aplicación web que una vez implantada, y sin necesidad de instalar software adicional permita:

- Al alumno:
 - Poder contactar por videoconferencia con su tutor docente, para ello simplemente necesitará un navegador relativamente reciente con Adobe Flash Plug-in instalado.
Tanto el alumno como el docente también dispondrán en la sala virtual en la que se reúnan, una pizarra en la podrán escribir ambos de forma colaborativa y también un chat por si surgiera algún problema con el audio, etc,...
 - Poder anotar diariamente su ficha de seguimiento y generar un impreso en formato PDF semanalmente de su actividad diaria, con el formato y estructura que se puede ver a continuación en el gráfico 1:

Gráfico 1.- Modelo de ficha de seguimiento semanal

JUNTA DE ANDALUCÍA

CONSEJERÍA DE EDUCACIÓN

(Hoja ____ de ____)

FORMACIÓN EN CENTROS DE TRABAJO. FICHA SEMANAL DEL ALUMNO/ALUMNA

Semana del al de de 20.....	
CENTRO DOCENTE: PROFESOR/PROFESORA RESPONSABLE SEGUIMIENTO:	CENTRO DE TRABAJO COLABORADOR: TUTOR/TUTORA DEL CENTRO DE TRABAJO:
ALUMNO/ALUMNA:	CICLO FORMATIVO: GRADO: S

DÍA	ACTIVIDAD DESARROLLADA/PUESTO FORMATIVO	TIEMPO EMPLEADO	OBSERVACIONES
LUNES			
MARTES			
MIÉRCOLES			
JUEVES			
VIERNES			

EL/LA ALUMNO/A

Vº Bº EL/LA PROFESOR/A RESPONSABLE
DEL SEGUIMIENTO

Vº Bº EL/LA TUTOR/A DEL CENTRO DE TRABAJO

Fdo.: _____

Fdo.: _____

Fdo.: _____



- Al tutor docente:
 - Poder tele-visitarse al alumno y/o tutor laboral: Será el tutor docente el que podrá invitar a los alumnos y/o tutores laborales a su sala. Quiere esto decir que cada tutor docente dispondrá de su propia sala virtual. Además de estas salas dedicadas, existe una sala común donde todos los usuarios tendrán acceso libre.
 - Poder revisar las anotaciones diarias que realizan sus alumnos.
 - Dar de alta a las empresas participantes, así como sus centros de trabajo asociados y tutores laborales colaboradores.
 - Crear la ficha de programa formativo de sus alumnos. Gráfico 2.
 - Crear los contratos y generar documentación de empresa colaboradora y los alumnos, en formato PDF para su posterior impresión y entrega en Secretaría, quien será la encargada de formalizar dichos contratos. Gráfico 3.
 - Poder anotar las visitas realizadas a las empresas y poder generar dicho informe. Gráfico 4.

CONSEJERÍA DE EDUCACIÓN

FORMACIÓN EN CENTROS DE TRABAJO. PROGRAMA FORMATIVO

(Hoja ... de ...)

CENTRO DOCENTE: PROFESOR/PROFESORA RESPONSABLE DEL SEGUIMIENTO:	CENTRO DE TRABAJO COLABORADOR: TUTOR / TUTORA DEL CENTRO DE TRABAJO:	
ALUMNO/A ALUMNA:	PERÍODO DE REALIZACIÓN DE LA FCT <input type="checkbox"/> PRIMER TRIMESTRE <input type="checkbox"/> SEGUNDO TRIMESTRE <input type="checkbox"/> TERCER TRIMESTRE	
CURSO ESCOLAR: _____ / _____		
FAMILIA PROFESIONAL:	CICLO FORMATIVO:	GRADO:
<i>RESULTADOS DE APRENDIZAJE</i>	<i>ACTIVIDADES FORMATIVO-PRODUCTIVAS</i>	<i>CRITERIOS DE EVALUACIÓN</i>

En _____ a _____ de _____ de 2.0 _____

EL/LA PROFESOR/A RESPONSABLE DEL SEGUIMIENTO

EL/LA JEFE/A DEL DEPARTAMENTO DE FAMILIA
PROFESIONAL

EL/LA RESPONSABLE DEL CENTRO DE TRABAJO

Edo:

Edo:

Gráfico 3.- Modelo de documentación empresa colaboradora

FORMACIÓN EN CENTROS DE TRABAJO

CICLO FORMATIVO ADMINISTRACIÓN DE SISTEMAS INFORMÁTICOS

DATOS DE LA EMPRESA

RAZON SOCIAL: Zalt - EMPRESA DE PRUEBA, SL DIRECCION: Rto. Interior Zoma Franca POBLACION: Cádiz TELEFONO: 956897845 E-MAIL: tedeum@tedeum.com GERENTE D. Sebas Jiménez Sánchez TUTOR LABORAL D. TUTOR LABORAL DE PRUEBA TUTOR DOCENTE D. TUTOR DOCENTE DE PRUEBA	CIF: B11557618 C. POSTAL: 11007 FAX: 956784512 DNI: 31258741Q DNI: 31258742V DNI: 123456Z
---	--

CENTRO DE TRABAJO

DIRECCION: Rto. Interior Zona Franca
POBLACION: Cádiz **C. POSTAL:** 11007
TELEFONO: 956897845 **FAX:** 956784512 **E-MAIL:** tedeum@tedeum.com

DATOS ALUMNO

APELLIDOS Y NOMBRE: Z. Alumno de Prueba 2
DNI: 6647 1258Q **TELÉFONO:** 9569 569 56 **MOVIL:** 65 465 465 4
FECHA DE NACIMIENTO: 1 de enero de 1948 **MODALIDAD DE AYUDA ¹:** B

ⁱ A. Misma localidad desde 5 Kms
B. Otra localidad hasta 20 Kms
C. Otra localidad a más de 20 Kms

Gráfico 4.- Modelo de visita a empresa

HOJA DE VISITA A EMPRESA

Curso 2012/2013

Tutor/a Docente	D/Dº TUTOR D.OCENTE DE PRUEBA
Alumno/a	D/Dº Alumno de Prueba 2 Z
Ciclo Formativo	Administración de Sistemas Informáticos en Red
Tutor/a Laboral	TUTOR LABORAL DE PRUEBA
Empresa	Zalt - EMPRESA DE PRUEBA, SL
Dirección	Rto. Interior Zona Franca
Población	Cádiz
Código postal	11007

1. La visita se realiza para evaluar si el alumno/a

1.1.	Identifica la estructura y organización de la empresa relacionándolas con el tipo de servicio que presta.
1.2.	Aplica hábitos éticos y laborales en el desarrollo de su actividad profesional de acuerdo con las características del puesto de trabajo y con los procedimientos establecidos en la empresa.
1.3.	Organiza el procedimiento de trabajo que debe desarrollar, interpretando la documentación específica.
1.4.	Determina las características técnicas de la instalación a partir de las funcionalidades y necesidades establecidas.
1.5.	Participa en el diseño, la puesta en marcha y el mantenimiento de instalaciones con servicios de red local e Internet, documentando la intervención realizada.
1.6.	Asiste a los usuarios resolviendo problemas de la explotación del sistema, según las normas y tiempos establecidos.

2. Si el motivo de la visita es otro o se quiere constatar alguna observación, indicar cuál

3. Para realizar esta visita se necesitó 1 hora computable como lectiva

En Cádiz a 3 de junio de 2013

Sello
de la
empresa

El/la Tutor/a Laboral

El/la Tutor/a Docente

Fdo.- TUTOR LABORAL DE PRUEBA

Fdo.- TUTOR D.OCENTE DE PRUEBA

MODALIDAD VISITA

X Presencial

Telemática

- Al tutor laboral:
 - Poder contactar por videoconferencia con el tutor docente de cada alumno.
 - Poder revisar las anotaciones diarias que realizan todos sus alumnos.

1.4.- Planteamiento y evaluación de diversas soluciones

Partiendo de la base que lo que quería desarrollar era una aplicación web y que el lenguaje de programación y framework estudiado en el ciclo formativo ha sido Python/Django, la primera de las posibles soluciones era para mí de elección inmediata, ya que me interesaba profundizar más en el desarrollo con este framework.

Django es un framework web de código abierto escrito en Python que permite construir aplicaciones web más rápido y con menos código. Cuenta con una abundante y excelente documentación así como una amplia comunidad de desarrolladores.

Salvado el primer obstáculo lo que en primer lugar traté de solucionar fue de qué forma se llevaría a cabo la videoconferencia teniendo en cuenta las siguientes premisas:

- Sólo sería necesario disponer de un navegador web relativamente reciente, siendo indiferente que éste se ejecutase sobre Windows, GNU/Linux o Mac.
- No será necesario instalar nada más: Este punto actualmente no es estrictamente cierto, porque el navegador web deberá disponer del Plug-in de Adobe Flash Player. De todas formas es una tecnología omnipresente. Muchísimas páginas web contienen animaciones, menús y botones que requieren Adobe Flash Player, lo cual lo convierte en un Plug-in imprescindible, haciendo que casi la totalidad de los usuarios lo tengan ya instalado en su navegador.

Teniendo en cuenta las anteriores premisas y dando por hecho que tendría que apoyarme en una solución de terceros encontré Opentok y Google Talk.

En principio la 2ª opción me atraía bastante por:

- Poder utilizarlo como parte del entorno de desarrollo de Google (Google App Engine - GAE), que además permite desarrollar en varios lenguajes de programación, entre ellos Python/Django.
- Alojar directamente la aplicación en los servidores de Google y con costes muy reducidos.

Tras varias semanas experimentando y documentándome sobre el uso de GAE, decidí descartarla por varios motivos:

- GAE no permitía instalar nada. Esto hizo plantearme de inmediato que necesitaría un servidor dedicado para mi aplicación que me permitiese instalar lo que me hiciese falta,...
- GAE utiliza un sistema de almacenamiento no relacional muy interesante, escalable, etc,... pero que me haría la migración a otro sistema bastante complicada, ya que tendría que reescribir por completo todo el código referente al Modelo de la aplicación.
- Para la autenticación de los usuarios, sería prácticamente necesario disponer de una cuenta de Gmail. Esto es visto por mucha gente como una limitación.
- Lo que hizo descartarla del todo es que justo en esas semanas de documentación y pruebas, Google hizo bastantes cambios en la API de Google Talk (marzo-mayo de 2012), lo cual desencadenó en numerosas protestas de desarrolladores,

- etc,...actualmente Google Talk se denomina Google Hangouts,...
- Para hacer uso de Google Talk es necesario descargar e instalar un Plug-in en el navegador que prácticamente nadie tiene instalado.

La opción elegida por tanto para desarrollar el tema de la videoconferencia ha sido [Opentok](#), el cual permite a desarrolladores integrar videoconferencia en sus aplicaciones escritas en PHP, Python, Java, .NET o Ruby, proporcionando un control total sobre la selección, el diseño y la interacción con los distintos streams de audio y video.

El uso de Opentok no está al 100% libre de costes, aunque para el uso al que va a ser destinado la aplicación cumple de sobra, ya que ofrece de manera gratuita:

- Minutos ilimitados en videoconferencias 1:1
- Hasta 25 mil minutos/mes en videoconferencias de grupo, es decir 3 o más participantes.

Opentok posee una amplia comunidad de desarrolladores, una buena documentación y una curva de aprendizaje muy rápida.

Siguiendo con las tecnologías utilizadas, decir que como SGBD he elegido PostgreSQL, ya que además de ser Open-Source, potente, estable, extensible, multiplataforma, contar con procedimientos almacenados, triggers, etc,... es el SGBD recomendado por la comunidad Django, existiendo en el ORM de Django características exclusivas para PostgreSQL.

Sobra decir los motivos por los que he elegido Gnu/Linux, en concreto Ubuntu Server 12.04 como Sistema Operativo.

1.5.- Justificación de la solución elegida

La base para la elección de las diferentes tecnologías utilizadas ha sido sobre todo:

- Mantener la filosofía del software libre en la que se ha hecho especial hincapié a lo largo de todo el ciclo formativo.
- Construir una aplicación multiplataforma.
- El hecho de elegir una aplicación web es la fácil implantación de dicha solución, no requiriendo la instalación de software adicional para los usuarios en sus equipos.

1.6.- Modelado de la solución

El proyecto se ha dividido en 2 aplicaciones claramente diferenciadas, por un lado he separado todo lo relativo al tema de la videoconferencia y por el otro lado está lo relativo a la FCT propiamente dicha.

Para plasmar dicha realidad y hacerme una clara idea de cómo funciona dicho sistema me hizo falta recurrir además de a mi tutor de proyecto, al profesor Jaime Canillas, el cual, por ser el que cuenta con más experiencia en el seguimiento de la FCT, me ha podido describir con amplio detalle cuáles son los agentes implicados, qué funciones e implicaciones tiene cada uno, etc... y partiendo de toda esa información recogida he intentado reflejar con el mayor detalle posible dicho sistema, el cual se puede completar, si se necesitara, con nuevas ideas, tareas, modificaciones o ampliaciones.

Partiendo de toda la documentación facilitada acerca del seguimiento de la FCT y con la ayuda y sugerencias del tutor de proyecto se estudiarán qué datos deben estar disponibles en el sistema, de tal forma que se permita realizar el mayor número de tareas, así como consultas de la información existente en nuestro sistema.

1.6.1.- Recursos humanos

Este proyecto se ha realizado de forma individual con el seguimiento y apoyo del tutor de proyecto, pero han colaborado durante todo el periodo de prácticas los alumnos y profesores de 2º de ASIR del curso 2012-2013, los cuales me han hecho llegar sus sugerencias y aportaciones para mejorar dicha aplicación. Ya que durante la primera y tercera evaluación se ha estado utilizando durante la fase de prácticas en empresas, por tanto agradecer desde aquí a los compañeros: **Manolo, Adri, Manu, Sara, Ana, Yasmina, Emilio, Alberto, Farru, Rocio** y a los profesores: **Jaime, Diego y Javier** por colaborar y hacerme llegar sus sugerencias y aportaciones.

1.6.2.- Recursos hardware

En un primer momento utilicé para pruebas y para desarrollar los primeros prototipos un equipo obsoleto que disponía en casa, el cual debido al excesivo ruido que generaba, y pensando en el incremento del consumo eléctrico que me supondría en la fase de pruebas con los alumnos y profesores al estar las 24 horas encendido, descarté dicha opción y me decidí por hacer uso de una micro-instancia de Ubuntu Server 12.04 64 bits de, [Amazon Elastic Compute Cloud \(Amazon EC2\)](#) gracias a la [capa de uso gratuito](#) que ofrece durante todo un año [Amazon Web Services \(AWS\)](#).

Amazon Elastic Compute Cloud (Amazon EC2) es un servicio web que proporciona capacidad informática con tamaño modificable en la nube. Está diseñado para facilitar a los desarrolladores recursos informáticos escalables basados en Web.

La sencilla interfaz de servicios web de Amazon EC2 permite obtener y configurar su capacidad con una fracción mínima. Proporciona un control completo sobre sus recursos informáticos y permite ejecutarse en el entorno informático acreditado de Amazon.

Amazon EC2 reduce el tiempo necesario para obtener y arrancar nuevas instancias de servidor en minutos, lo que permite escalar rápidamente la capacidad, ya sea aumentándola o reduciéndola, según cambien sus necesidades.

Amazon EC2 presenta un auténtico entorno informático virtual, que permite utilizar interfaces de servicio web para iniciar instancias con distintos sistemas operativos, cargarlas con su entorno de aplicaciones personalizadas, gestionar sus permisos de acceso a la red y ejecutar su imagen utilizando los sistemas que desee.

Capa de uso gratuito de AWS: AWS presenta una capa de uso gratuito. La capa de uso gratuito puede emplearse para todo aquello que desee realizar en la nube: ejecutar nuevas aplicaciones, probar aplicaciones existentes en la nube o simplemente obtener experiencia práctica con AWS.

El concreto el servicio que llevo usando de manera gratuita desde el 23 de julio de 2012 es Amazon EC2 (Amazon Elastic Compute Cloud), un servicio web que proporciona 750 horas de micro-instancias Linux (613 MB de memoria y compatibilidad con plataformas de 32 y 64 bits); horas suficientes para disfrutar cada mes de una ejecución constante al mes durante un año. 8 GB de espacio en disco y 15 GB de ancho de banda saliente en todos los servicios de AWS.

1.6.3.- Recursos software

Todo el software utilizado es Software Libre y gratuito, como he mencionado anteriormente, siendo uno de los requisitos fundamentales para su elección.

- Sistema Operativo: Ubuntu Server 12.04 64 bits
- Desarrollada con Django y utilizando algunos módulos Python
- Como servidor web Apache
- SGBD: PostgreSQL. Para administrarla de forma gráfica he utilizado phppgadmin
- Para posibilitar la videoconferencia he utilizado la API Python y Javascript de Opentok.
- NodeJS + Etherpad para posibilitar la edición de un documento colaborativo en la aplicación de tele-visita.

1.7.- Planificación temporal

Para el desarrollo del presente proyecto se han dispuesto formalmente un total de 11 semanas, coincidiendo con el periodo de la FCT en la 3ª evaluación del curso 2012-2013, no obstante, el inicio del proyecto se inició en el mes de abril de 2012, fecha desde la cual y hasta el mes de junio del mismo año elaboré un primer prototipo que cubría más bien la aplicación de videoconferencia y algo de lo referente a la documentación de la FCT.

Posteriormente en septiembre de 2012, coincidiendo con el inicio del curso 2012-2013 y estando ya la aplicación alojada en AWS, se pudo probar la aplicación con un alumno que realizó la FCT en la primera evaluación, aunque aún le faltaban muchas funcionalidades a la aplicación, sirvió para poder ir solucionando algunos problemas.

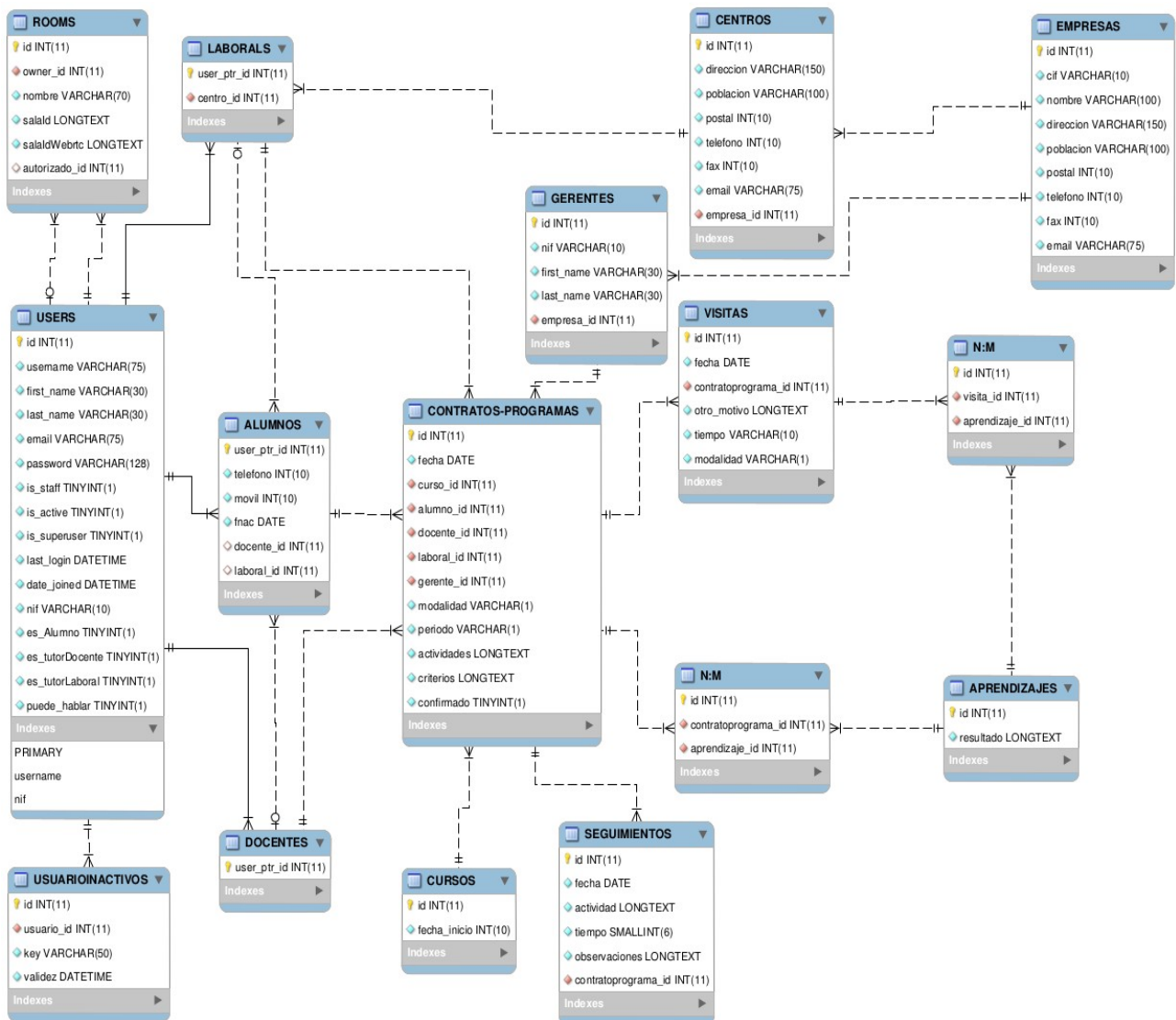
Finalmente al comienzo de la tercera evaluación, sobre el mes de abril de 2013 y coincidiendo que un buen grupo de alumnos realizarían las prácticas de empresa, fue cuando se desarrollaron la gran mayoría de funcionalidades.

La dos últimas semanas se emplearán en terminar las partes que hayan podido quedar pendientes, se finalizará la documentación del proyecto que se debe presentar, y se preparará la exposición y defensa del mismo.

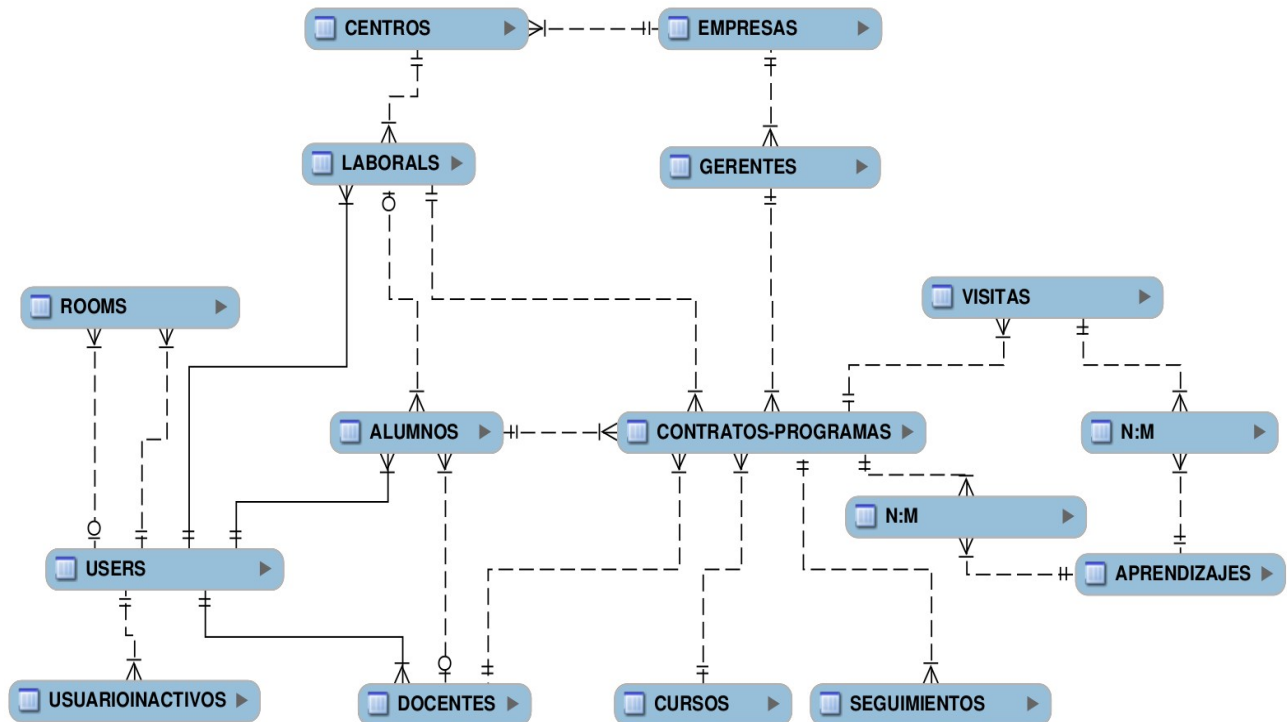
2.- EJECUCIÓN DEL PROYECTO

En primer lugar y antes de empezar a describir las partes del sistema voy a mostrar cómo voy a tratar de representar las entidades relevantes que lo componen así como sus interrelaciones y propiedades, es decir a continuación el Modelo de Datos Entidad-Relación (E/R) del proyecto.

2.1.- Modelo de Datos Entidad-Relación (E/R)



...eliminando los atributos para que se vea un poco mas claro.



En primer lugar señalar que en dicho sistema se ha incluido la entidad usuario (User), que es una entidad propia de Django que automáticamente crea y es común a todas las aplicaciones contenidas en dicho proyecto.

2.1.1.- La entidad USER

Columna	Tipo de dato	No Nulo
id	integer	NOT NULL
username	character varying(75)	NOT NULL
first_name	character varying(30)	NOT NULL
last_name	character varying(30)	NOT NULL
email	character varying(75)	NOT NULL
password	character varying(128)	NOT NULL
is_staff	boolean	NOT NULL
is_active	boolean	NOT NULL
is_superuser	boolean	NOT NULL
last_login	timestamp with time zone	NOT NULL
date_joined	timestamp with time zone	NOT NULL
nif	character varying(10)	NOT NULL
es_Alumno	boolean	NOT NULL
es_tutorDocente	boolean	NOT NULL
es_tutorLaboral	boolean	NOT NULL
puede_hablar	boolean	NOT NULL

Los atributos y tipos de datos de la entidad User que Django proporciona por defecto son los que aparecen en la imagen anterior pero sin sombrear, los que aparece sombreados son atributos que se pueden añadir a dicho modelo y vi conveniente añadir para controlar mejor algunas consultas.

Podría explicarse más adelante, pero para añadir atributos al modelo USER, simplemente hay que añadir al fichero **models.py** de la aplicación lo siguiente:

```
# -*- coding: utf-8 -*-
# model.py

from django.db import models
from django.contrib.auth.models import User

User.add_to_class('nif', models.CharField(max_length=10, unique=True))
User.add_to_class('es_Alumno', models.BooleanField(default=False))
User.add_to_class('es_tutorDocente', models.BooleanField(default=False))
User.add_to_class('es_tutorLaboral', models.BooleanField(default=False))
User.add_to_class('puede_hablar', models.BooleanField(default=False))
```

Django siempre nombra a las claves primarias (PK) como **id**. Comentar además que tanto el atributo **username** (de serie en el modelo User) como **nif** (añadido a la clase User,..) tienen la restricción UNIQUE.

Los tres tipos de usuarios que aparecen en el sistema y que por tanto heredan del modelo User de Django son ALUMNO, TUTOR_DOCENTE y TUTOR_LABORAL. Cada tipo de usuario formará una entidad independiente y sólo se le añadirá aquellos atributos que se necesite para la aplicación.

2.1.2.- ALUMNO, TUTOR_DOCENTE y TUTOR_LABORAL

```
#models.py
from django.db import models
from django.contrib.auth.models import User

class Laboral(User):
    centro = models.ForeignKey(Centro)

    def __unicode__(self):
        return u'%s %s' % (self.first_name, self.last_name)

class Docente(User):

    def __unicode__(self):
        return u'%s %s' % (self.first_name, self.last_name)

class Alumno(User):
    telefono = models.PositiveIntegerField(max_length=9)
    movil = models.PositiveIntegerField(max_length=9)
    fnac = models.DateField(verbose_name='Fecha de nacimiento')
    docente = models.ForeignKey(Docente, blank=True, null=True, on_delete=models.SET_NULL)
    laboral = models.ForeignKey(Laboral, blank=True, null=True, on_delete=models.SET_NULL)

    def __unicode__(self):
        return u'%s %s' % (self.first_name, self.last_name)
```

Por ejemplo a TUTOR_DOCENTE, no le he añadido ningún atributo más del que ya hereda de User, ya que el tutor docente de la aplicación está ligado firmemente al centro IES Fernando Aguilar Quignón y al ciclo de ASIR, tal y como detallé en el apartado “Alcance del proyecto” de mi anteproyecto. Lógicamente si quisiéramos extender dicho modelo para diferentes centros de enseñanza y ciclos formativos necesitaríamos crear nuevas entidades (Centros y Ciclos) y crear interrelaciones desde tutor docente mediante claves foráneas a dichas entidades.

A la entidad TUTOR_LABORAL sí le he añadido un atributo, que no es más que una clave foránea a la entidad CENTROS (Centro de trabajo), al cual está vinculado como norma general un trabajador. He obviado la posibilidad del pluriempleo en los tutores laborales, por no complicar el sistema con una interrelación N:M que veo innecesaria ya que generalmente la FCT se desarrolla en horario de mañana y dicho trabajador, suponiendo que fuese pluriempleado estará por la mañana en un sitio u otro....Cada CENTRO pertenecerá a una EMPRESA concreta, pudiendo tener una empresa uno o varios centros de trabajo. Los demás atributos tanto de CENTRO como de EMPRESA no revisten mayor importancia y son básicamente la localización y algunos datos identificativos básicos. Sólo destacar que el atributo **cif** de EMPRESA tiene la restricción UNIQUE.

2.1.3.- CENTRO y EMPRESA

```
#models.py
from django.db import models

class Centro(models.Model):
    direccion = models.CharField(max_length=150)
    poblacion = models.CharField(max_length=100)
    postal = models.PositiveIntegerField(max_length=5)
    telefono = models.PositiveIntegerField(max_length=9)
    fax = models.PositiveIntegerField(max_length=9)
    email = models.EmailField()
    empresa = models.ForeignKey(Empresa)

    def __unicode__(self):
        return u'%s: %s' % (self.poblacion, self.direccion)

class Empresa(models.Model):
    cif = models.CharField(max_length=10, unique=True)
    nombre = models.CharField(max_length=100, unique=True)
    direccion = models.CharField(max_length=150)
    poblacion = models.CharField(max_length=100)
    postal = models.PositiveIntegerField(max_length=5)
    telefono = models.PositiveIntegerField(max_length=9)
    fax = models.PositiveIntegerField(max_length=9)
    email = models.EmailField()

    def __unicode__(self):
        return self.nombre
```

Existe otra entidad, llamada GERENTE, que también está relacionada mediante una PK con la entidad EMPRESA. La entidad GERENTE no hereda nada del modelo USER, ya que no necesitará interactuar con la aplicación, es decir los GERENTES registrados en la aplicación no podrán autenticarse contra la aplicación, no siendo por tanto usuarios logueables. El hecho de su existencia no es más que por la necesidad de registrarlos para la posterior confección de los contratos de colaboración de la FCT, ya que podría darse la coincidencia (Pymes y autónomos) de que coincidan tanto el tutor laboral y el gerente, pero generalmente no es así. En caso de que coincidan, en la aplicación habrá que registrar a un tutor docente y a un gerente por separado, aún siendo la misma persona.

2.1.4.- GERENTE

```
#models.py
from django.db import models

class Gerente(models.Model):
    nif = models.CharField(max_length=10, unique=True)
    first_name = models.CharField(max_length=30)
    last_name = models.CharField(max_length=30)
    empresa = models.ForeignKey(Empresa)

    def __unicode__(self):
        return u'%s %s' % (self.first_name, self.last_name)
```

Antes de continuar mencionar que existe una tabla estática, es decir sólo se escribirá sobre ella una vez, es la tabla APRENDIZAJES, que corresponde con aquellos resultados de aprendizaje descritos en la ORDEN de 19 de julio de 2010, de la Consejería de Educación de Andalucía, donde se establece los resultados de aprendizaje para el módulo de la FCT del ciclo de ASIR. Dichos resultados de aprendizaje se encuentran detallados en dicha programación didáctica y son:

1_Identifica la estructura y organización de la empresa relacionándolas con el tipo de servicio que presta.

2_Aplica hábitos éticos y laborales en el desarrollo de su actividad profesional de acuerdo con las características del puesto de trabajo y con los procedimientos establecidos en la empresa.

3_Organiza el procedimiento de trabajo que debe desarrollar, interpretando la documentación específica.

4_Determina las características técnicas de la instalación a partir de las funcionalidades y necesidades establecidas.

5_Participa en el diseño, la puesta en marcha y el mantenimiento de instalaciones con servicios de red local e Internet, documentando la intervención realizada.

6_Asiste a los usuarios resolviendo problemas de la explotación del sistema, según las normas y tiempos establecidos.

2.1.5.- APRENDIZAJES

```
#models.py
from django.db import models

class Aprendizaje(models.Model):
    resultado = models.TextField(max_length=300)

    def __unicode__(self):
        return u'%s' % (self.resultado)
```

Esta tabla tras crearse, se insertarán los anteriores 6 registros automáticamente nada mas instalar la aplicación en un nuevo servidor.

Es importante que dicha tabla se encuentre ya poblada nada más instalar la aplicación, no sólo porque habría que estar pendiente de poblarla nada más empezar a utilizar la aplicación, sino porque de esta tabla dependen algunas otras tablas, como las que detallo a continuación.

La entidad CONTRATOS_PROGRAMAS es donde se crean los contratos y los programas formativos asociados a dicho contrato, cada registro como anteriormente podía verse está compuesto por una fecha de creación, un curso (FK a la entidad CURSOS), un alumno (FK a la entidad ALUMNOS), un docente (FK a la entidad TUTOR_DOCENTE), un t. laboral (FK a TUTOR_LABORAL) un gerente que deberá firmar el contrato (FK a la GERENTES), modalidad de desplazamiento, trimestre en que se realizará la FCT, un atributo tipo texto llamado actividades, otro de criterios de evaluación y por último el que está relacionado con la entidad anterior (APRENDIZAJES), que lo hará mediante una interrelación tipo N:M, ya que podrán vincularse con uno o más de los criterios indicados de aprendizajes deseados y detallados en el programa formativo del módulo.

2.1.6.- CONTRATOS_PROGRAMAS

```
#models.py
from django.db import models
import datetime

class ContratoPrograma(models.Model):
    MODALIDADES = (
        ('A', 'Misma localidad desde 5 Kms'),
        ('B', 'Otra localidad hasta 20 Kms'),
        ('C', 'Otra localidad a mas de 20 Kms'),
    )
    TRIMESTRES = (
        ('1', 'Primer Trimestre'),
        ('2', 'Segundo Trimestre'),
        ('3', 'Tercer Trimestre'),
    )
    fecha = models.DateField(default=datetime.date.today())
    curso = models.ForeignKey(Curso)
    alumno = models.ForeignKey(Alumno)
    docente = models.ForeignKey(Docente)
    laboral = models.ForeignKey(Laboral)
    gerente = models.ForeignKey(Gerente)
    modalidad = models.CharField(max_length=1, choices=MODALIDADES)
    periodo = models.CharField(max_length=1, choices=TRIMESTRES)
    aprendizajes = models.ManyToManyField(Aprendizaje)
    actividades = models.TextField(max_length=1000, blank=True)
    criterios = models.TextField(max_length=1000, blank=True)
    confirmado = models.BooleanField(default=False)

    def __unicode__(self):
        return u'Curso: %s Alumno: %s Empresa: %s' % (self.curso,
            self.alumno, self.gerente.empresa)
```

La anterior entidad hace referencia, entre otras, a la tabla CURSOS, sobre esta tabla insertará registros directa y automáticamente la propia aplicación, tan sólo consta de un campo llamado fecha_inicio, donde se guardará el año de inicio del curso.
CURSOS en el modelo de Django:

```
#models.py
from django.db import models

class Curso(models.Model):
    fecha_inicio = models.PositiveIntegerField(max_length=4, unique=True)

    def __unicode__(self):
        return u'%s/%s' % (self.fecha_inicio, self.fecha_inicio + 1)
```

Paso a continuación a describir 2 entidades sobre las que se crearán el mayor número de registros, son las tablas SEGUIMIENTOS y VISITAS.

En la tabla SEGUIMIENTOS será donde los alumnos irán insertando aquellos registros referentes a las actividades diarias que realizan en las prácticas de empresas y que podrán ser también consultadas posteriormente tanto por su tutor docente y/o laboral. El alumno también podrá generar un documento en formato PDF por cada semana de prácticas.

2.1.7.- SEGUIMIENTOS

```
#models.py
from django.db import models

class Seguimiento(models.Model):
    fecha = models.DateField()
    actividad = models.TextField(max_length=200)
    tiempo = models.SmallIntegerField()
    observaciones = models.TextField(max_length=200, blank=True)
    contratoprograma = models.ForeignKey(ContratoPrograma)

    def __unicode__(self):
        return u'%s %s %s' % (self.fecha, self.contratoprograma.alumno, self.contratoprograma.gerente.empresa)
```

Para la insertar un registro en dicha tabla por el alumno, aparte de las limitaciones propias como tipo y longitud de los datos he añadido 3 más que son:

- No podrá crearse un registro con fecha posterior a la actual.
- Tampoco anterior a 90 días a la fecha actual. Por entenderse que habría dado tiempo ya de sobra para ir rellenando la ficha diaria y además para evitar que por error, como ha ocurrido en la fase de pruebas, un alumno inserte un registro de por ejemplo el año 2002, cuando la intención era escribir el año 2012. En principio se dispone en el formulario de registro de tareas diarias de un desplegable tipo calendario que evitaría esto, aunque también es cierto que no impide al alumno escribir manualmente la fecha en dicho campo, de ahí el error.
- La última limitación que se me ocurrió añadir es comprobar primero que la fecha insertada corresponde con un día de lunes a viernes.

Estas restricciones las controlo con validaciones del lado del servidor, contenidas en el **forms.py** de la aplicación, consiste en añadir a la propia clase formulario que controla la introducción de un registro de SEGUIMIENTO, una función con el formato **clean_** y el campo que quieres validar:

```
def clean_fecha(self):
    fecha = self.cleaned_data['fecha']
    diasemana = fecha.isoweekday()
    meses_dif = timedelta(days=90)
    if fecha > date.today():
        raise forms.ValidationError("Todavía no hemos llegado a esa fecha")
    elif (fecha + meses_dif) < date.today():
        raise forms.ValidationError("No puedes seleccionar una fecha con una antigüedad superior a 90 días")
    elif diasemana == 6 or diasemana == 7:
        raise forms.ValidationError("Tienes que seleccionar un día de lunes a viernes")

    return fecha
```

La entidad VISITAS, será utilizada en exclusiva por el tutor docente y le será de utilidad para anotar las visitas realizadas a los alumnos en la empresa donde realicen las practicas, bien sea de forma presencial o en la modalidad de videoconferencia o tele-visita. Podrá también generar un documento en formato PDF que tras la firma y sello de la empresa colaboradora le servirá al tutor docente para justificar ante el centro educativo el tiempo e inversión económica derivados de dicha visita.

Cada visita tendrá por objetivo vigilar si el alumno cumple alguno/s de los resultados de aprendizaje esperados (RAE) que se seleccionaron en el momento de la creación del programa formativo de la FCT del alumno al que se le va a realizar la visita. No será preciso por tanto seleccionar que se van a evaluar todos los RAE descritos en el programa. Estos RAE objeto de evaluación en la visita, serán registrados en una tabla intermedia entre la entidad VISITAS y APRENDIZAJES, puesto que es una interrelación N:M.

2.1.8.- VISITAS

```
#models.py
from django.db import models

class Visita(models.Model):
    MODALIDADES = (
        ('P', 'Presencial'),
        ('T', 'Telematica'),
    )
    fecha = models.DateField(default=datetime.date.today())
    contratoprograma = models.ForeignKey(ContratoPrograma)
    aprendizajesPositivos = models.ManyToManyField(Aprendizaje, blank=True)
    otro_motivo = models.TextField(max_length=130, blank=True)
    tiempo = models.CharField(max_length=10)
    modalidad = models.CharField(max_length=1, choices=MODALIDADES)

    def __unicode__(self):
        return u'Fecha: %s Alumno: %s Empresa: %s' % (self.fecha, self.contratoprograma.alumno,
                                                    self.contratoprograma.gerente.empresa)
```

2.1.9.- USUARIOS-INACTIVOS

Llegado ha este punto he de explicar cómo he manejado el tema del registro y autenticación de los usuarios. Esta aplicación no es la clásica aplicación web en la que el usuario rellena un formulario de registro y una vez validados dichos datos, éste ya queda registrado, sino que en este caso los usuarios son invitados por un tercero.

Desde un punto de vista practico, el único usuario que se auto-registra es el administrador en el momento de instalar/implantar la aplicación, por tanto los DOCENTES serán los primeros usuarios que tendrá que dar de alta el administrador, los cuales *recibirán un email con las instrucciones para activar su cuenta de usuario y definir su password*. Este email junto con otros datos han sido previamente introducidos en la aplicación por el administrador mediante un formulario de registro.

Luego serán los DOCENTES quienes registrarán y en cierto modo invitaran a los ALUMNOS y TUTORES LABORALES a activar sus cuentas previamente creadas por él. Éstos igualmente recibirán un correo para activar su cuenta y definir su password y un nuevo correo de confirmación.

Entonces la entidad USUARIOSINACTIVOS es una tabla que temporalmente almacena a aquellos usuarios que han sido registrados pero los cuales aún no han activado su cuenta, en el momento que ellos la activen, pulsando en el enlace proporcionado en el correo, ese registro se elimina y el usuario queda activado.

La aplicación comprobará que coincide la clave alfanumérica aleatoria de 50 caracteres almacenada en la base de datos con aquella que se ha enviado al usuario y también comprueba que no han pasado mas días de los indicados en la constante definida en el fichero **settings.py** para activar a los usuarios.

```
#models.py
from django.db import models
import datetime
from django.contrib.auth.models import User
from django.conf import settings

class UsuarioInactivo(models.Model):
    usuario = models.ForeignKey(User, unique=True, verbose_name='Usuario a activar')
    key = models.CharField('Clave de activación', max_length=50)
    validez = models.DateTimeField(default=datetime.datetime.now() +
                                   datetime.timedelta(days=settings.DIAS_PARA_ACTIVAR_CUENTA))

#settings.py
DIAS_PARA_ACTIVAR_CUENTA = 2
```

La aplicación realizará una serie de comprobaciones:

- Que el NIF, email, etc,.. tengan una estructura válida
- Que no exista un usuario con el mismo NIF.

Al haber elegido como **username** el email, la aplicación comprobará que no exista ningún usuario con el mismo email y por tanto con el mismo username.

En un primer momento el haber elegido el campo email para asignárselo automáticamente al **username** me fue bien, pero rápidamente empecé a tener problemas, ya que por defecto Django (hasta la versión 1.4) utiliza para el **username** una longitud máxima de 30 caracteres y para el email de 75. Por ejemplo los correos electrónicos de los docentes van seguidos de **@iesfernandoaguiar.es** lo cual por sí sólo son 22 caracteres, al añadirle el nombre.apellido/s del docente ya son más de 30 caracteres.

Existen varias soluciones para este problema, entre ellas actualizar a la versión 1.5 de Django que solventa, entre muchos otros, este problema, pero lo descarté porque de la versión 1.3 a la 1.5 introdujeron modificaciones significativas, ya que fue un “salto grande” de versión y me obligaba a modificar bastante código, así que me decanté por una de las otras muchas soluciones. Lo que hice fue descargar otra aplicación Django llamada **fixusername** que no hace mas que eso (parchear el username), de ahí que en el modelo aparezca el atributo username con 75 caracteres.

Para instalarla solamente hubo que añadirla por delante de todas las aplicaciones en ese momento instaladas editando el fichero de configuración del proyecto settings.py:

```
INSTALLED_APPS = (  
    'misitio.fixusername',  
    'django.contrib.auth',  
    'django.contrib.contenttypes',  
    'django.contrib.sessions',  
    'django.contrib.staticfiles',  
    'misitio.presence',  
    'misitio.televisita',  
    'django.contrib.admin',  
)
```

Al tener PostgreSQL como SGBD, Django genera el siguiente DDL:

2.1.10.- DDL de la aplicación

```
BEGIN;  
CREATE TABLE "presence_empresa" (  
    "id" serial NOT NULL PRIMARY KEY,  
    "cif" varchar(10) NOT NULL UNIQUE,  
    "nombre" varchar(100) NOT NULL UNIQUE,  
    "direccion" varchar(150) NOT NULL,  
    "poblacion" varchar(100) NOT NULL,  
    "postal" integer CHECK ("postal" >= 0) NOT NULL,  
    "telefono" integer CHECK ("telefono" >= 0) NOT NULL,  
    "fax" integer CHECK ("fax" >= 0) NOT NULL,  
    "email" varchar(75) NOT NULL  
)  
;  
  
CREATE TABLE "presence_gerente" (  
    "id" serial NOT NULL PRIMARY KEY,  
    "nif" varchar(10) NOT NULL UNIQUE,  
    "first_name" varchar(30) NOT NULL,  
    "last_name" varchar(30) NOT NULL,  
    "empresa_id" integer NOT NULL REFERENCES "presence_empresa" ("id") DEFERRABLE INITIALLY DEFERRED  
)  
;  
  
CREATE TABLE "presence_centro" (  
    "id" serial NOT NULL PRIMARY KEY,  
    "direccion" varchar(150) NOT NULL,  
    "poblacion" varchar(100) NOT NULL,  
    "postal" integer CHECK ("postal" >= 0) NOT NULL,  
    "telefono" integer CHECK ("telefono" >= 0) NOT NULL,  
    "fax" integer CHECK ("fax" >= 0) NOT NULL,  
    "email" varchar(75) NOT NULL,  
    "empresa_id" integer NOT NULL REFERENCES "presence_empresa" ("id") DEFERRABLE INITIALLY DEFERRED  
)  
;  
  
CREATE TABLE "presence_laboral" (  
    "user_ptr_id" integer NOT NULL PRIMARY KEY REFERENCES "auth_user" ("id") DEFERRABLE INITIALLY DEFERRED,  
    "centro_id" integer NOT NULL REFERENCES "presence_centro" ("id") DEFERRABLE INITIALLY DEFERRED  
)  
;
```

```

CREATE TABLE "presence_docente" (
  "user_ptr_id" integer NOT NULL PRIMARY KEY REFERENCES "auth_user" ("id") DEFERRABLE INITIALLY DEFERRED
);

CREATE TABLE "presence_alumno" (
  "user_ptr_id" integer NOT NULL PRIMARY KEY REFERENCES "auth_user" ("id") DEFERRABLE INITIALLY DEFERRED,
  "telefono" integer CHECK ("telefono" >= 0) NOT NULL,
  "movil" integer CHECK ("movil" >= 0) NOT NULL,
  "fnac" date NOT NULL,
  "docente_id" integer REFERENCES "presence_docente" ("user_ptr_id") DEFERRABLE INITIALLY DEFERRED,
  "laboral_id" integer REFERENCES "presence_laboral" ("user_ptr_id") DEFERRABLE INITIALLY DEFERRED
);

CREATE TABLE "presence_curso" (
  "id" serial NOT NULL PRIMARY KEY,
  "fecha_inicio" integer CHECK ("fecha_inicio" >= 0) NOT NULL UNIQUE
);

CREATE TABLE "presence_aprendizaje" (
  "id" serial NOT NULL PRIMARY KEY,
  "resultado" text NOT NULL
);

CREATE TABLE "presence_contratoprograma_aprendizajes" (
  "id" serial NOT NULL PRIMARY KEY,
  "contratoprograma_id" integer NOT NULL,
  "aprendizaje_id" integer NOT NULL REFERENCES "presence_aprendizaje" ("id") DEFERRABLE INITIALLY DEFERRED,
  UNIQUE ("contratoprograma_id", "aprendizaje_id")
);

CREATE TABLE "presence_contratoprograma" (
  "id" serial NOT NULL PRIMARY KEY,
  "fecha" date NOT NULL,
  "curso_id" integer NOT NULL REFERENCES "presence_curso" ("id") DEFERRABLE INITIALLY DEFERRED,
  "alumno_id" integer NOT NULL REFERENCES "presence_alumno" ("user_ptr_id") DEFERRABLE INITIALLY DEFERRED,
  "docente_id" integer NOT NULL REFERENCES "presence_docente" ("user_ptr_id") DEFERRABLE INITIALLY DEFERRED,
  "laboral_id" integer NOT NULL REFERENCES "presence_laboral" ("user_ptr_id") DEFERRABLE INITIALLY DEFERRED,
  "gerente_id" integer NOT NULL REFERENCES "presence_gerente" ("id") DEFERRABLE INITIALLY DEFERRED,
  "modalidad" varchar(1) NOT NULL,
  "periodo" varchar(1) NOT NULL,
  "actividades" text NOT NULL,
  "criterios" text NOT NULL,
  "confirmado" boolean NOT NULL
);

ALTER TABLE "presence_contratoprograma_aprendizajes" ADD CONSTRAINT "contratoprograma_id_refs_id_9ad5dc3" FOREIGN
KEY ("contratoprograma_id") REFERENCES "presence_contratoprograma" ("id") DEFERRABLE INITIALLY DEFERRED;

```

```

CREATE TABLE "presence_seguimiento" (
  "id" serial NOT NULL PRIMARY KEY,
  "fecha" date NOT NULL,
  "actividad" text NOT NULL,
  "tiempo" smallint NOT NULL,
  "observaciones" text NOT NULL,
  "contratoprograma_id" integer NOT NULL REFERENCES "presence_contratoprograma" ("id") DEFERRABLE INITIALLY
DEFERRED
);

CREATE TABLE "presence_visita_aprendizajesPositivos" (
  "id" serial NOT NULL PRIMARY KEY,
  "visita_id" integer NOT NULL,
  "aprendizaje_id" integer NOT NULL REFERENCES "presence_aprendizaje" ("id") DEFERRABLE INITIALLY DEFERRED,
  UNIQUE ("visita_id", "aprendizaje_id")
);

CREATE TABLE "presence_visita" (
  "id" serial NOT NULL PRIMARY KEY,
  "fecha" date NOT NULL,
  "contratoprograma_id" integer NOT NULL REFERENCES "presence_contratoprograma" ("id") DEFERRABLE INITIALLY
DEFERRED,
  "otro_motivo" text NOT NULL,
  "tiempo" varchar(10) NOT NULL,
  "modalidad" varchar(1) NOT NULL
);

ALTER TABLE "presence_visita_aprendizajesPositivos" ADD CONSTRAINT "visita_id_refs_id_5cdea502" FOREIGN KEY ("visita_id")
REFERENCES "presence_visita" ("id") DEFERRABLE INITIALLY DEFERRED;

CREATE TABLE "presence_usuarioinactivo" (
  "id" serial NOT NULL PRIMARY KEY,
  "usuario_id" integer NOT NULL UNIQUE REFERENCES "auth_user" ("id") DEFERRABLE INITIALLY DEFERRED,
  "key" varchar(50) NOT NULL,
  "validez" timestamp with time zone NOT NULL
);

CREATE INDEX "presence_gerente_empresa_id" ON "presence_gerente" ("empresa_id");
CREATE INDEX "presence_centro_empresa_id" ON "presence_centro" ("empresa_id");
CREATE INDEX "presence_laboral_centro_id" ON "presence_laboral" ("centro_id");
CREATE INDEX "presence_alumno_docente_id" ON "presence_alumno" ("docente_id");
CREATE INDEX "presence_alumno_laboral_id" ON "presence_alumno" ("laboral_id");
CREATE INDEX "presence_contratoprograma_curso_id" ON "presence_contratoprograma" ("curso_id");
CREATE INDEX "presence_contratoprograma_alumno_id" ON "presence_contratoprograma" ("alumno_id");
CREATE INDEX "presence_contratoprograma_docente_id" ON "presence_contratoprograma" ("docente_id");
CREATE INDEX "presence_contratoprograma_laboral_id" ON "presence_contratoprograma" ("laboral_id");
CREATE INDEX "presence_contratoprograma_gerente_id" ON "presence_contratoprograma" ("gerente_id");
CREATE INDEX "presence_seguimiento_contratoprograma_id" ON "presence_seguimiento" ("contratoprograma_id");
CREATE INDEX "presence_visita_contratoprograma_id" ON "presence_visita" ("contratoprograma_id");
COMMIT;

```


Todo lo anteriormente expuesto sobre el modelo de mi proyecto se corresponde con una de las dos aplicaciones que la forman, en un principio el proyecto constaba a su vez de una aplicación, la cual abarcaba todo, incluido la parte de videoconferencia. Para que fuese más fácil modificar lo relativo a la videoconferencia, en caso de cambiar el Opentok por cualquier otro y también para hacer el proyecto más modular, decidí separarlo.

Justo cuando andaba pensando en separar el proyecto en dos aplicaciones, apareció la siguiente noticia en algunos medios: [Telefónica compra la compañía de videochat TokBox](#). Lo cual no hizo más que confirmarme en mi idea. A fecha de hoy, Telefónica, asegura que va a mantener vivo el proyecto Opentok y con la misma filosofía.

2.2.- Aplicación Tele-presencia

El modelo de datos utilizado para la aplicación videoconferencia es bastante sencillo y sólo consta de una entidad, llamada ROOM, y no son más que salas de videoconferencia las cuales tienen un dueño. Este dueño y dado que sólo podrán crear salas los tutores docentes, recibirá el nombre de dicho docente. Podrán invitar a dichas salas a otros tutores docentes, a tutores laborales o a alumnos.

2.2.1.- ROOMS

```
#models.py
from django.db import models
from django.contrib.auth.models import User

class Room(models.Model):
    owner = models.ForeignKey(User, unique=True, related_name='owner')
    nombre = models.CharField(max_length=70)
    salaId = models.TextField(blank=True)
    salaIdWebRTC = models.TextField(blank=True)
    autorizado = models.ForeignKey(User, blank=True, null=True,
    on_delete=models.SET_NULL, related_name='autorizado')

    def __unicode__(self):
        return self.nombre
```

2.2.2.- DDL tele-visita

```
BEGIN;
CREATE TABLE "televisita_room" (
    "id" serial NOT NULL PRIMARY KEY,
    "owner_id" integer NOT NULL UNIQUE REFERENCES "auth_user" ("id") DEFERRABLE INITIALLY DEFERRED,
    "nombre" varchar(70) NOT NULL,
    "salaId" text NOT NULL,
    "salaIdWebRTC" text NOT NULL,
    "autorizado_id" integer REFERENCES "auth_user" ("id") DEFERRABLE INITIALLY DEFERRED
);

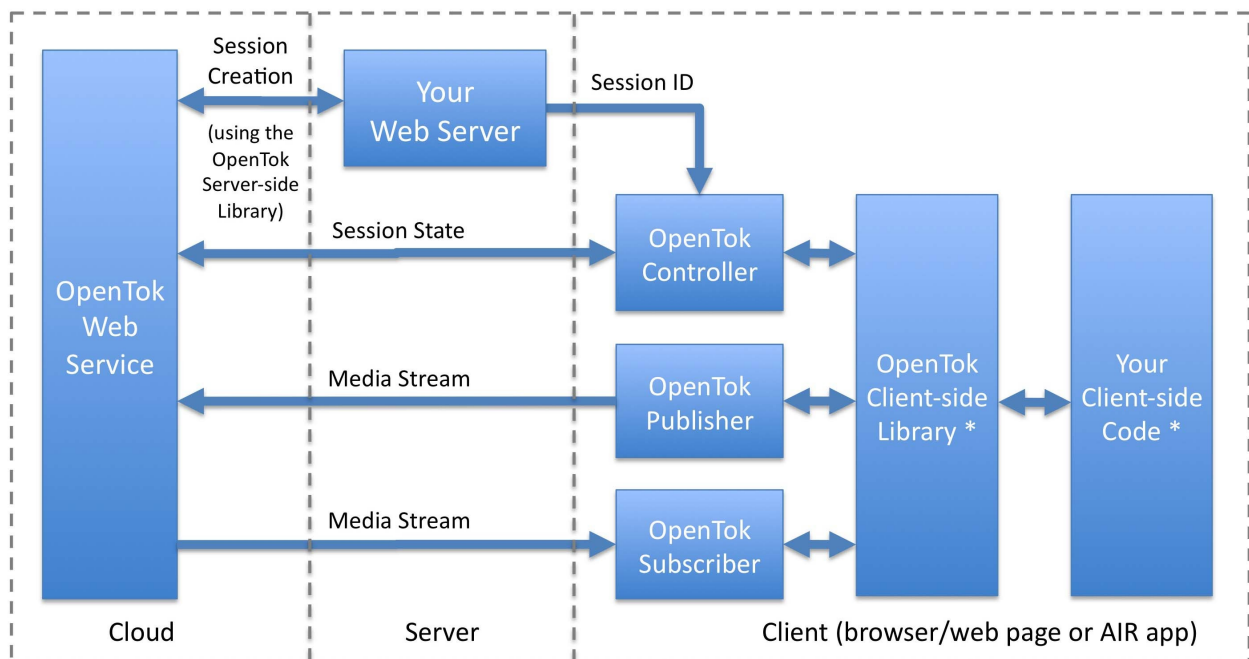
CREATE INDEX "televisita_room_autorizado_id" ON "televisita_room" ("autorizado_id");
COMMIT;
```

Opentok trabaja sobre el concepto de sesiones y tokens, donde una sesión es entendida como una sala y para cada sala ofrece la posibilidad de generar tokens, que no son mas que llaves para acceder a esas salas.

Para encajar estas piezas, Opentok ofrece 2 APIs para los desarrolladores:

- Una API del lado del servidor, escrita en varios lenguajes como Java, Ruby, etc,...yo he elegido la API Python y la he añadido a mi aplicación de tele-visita como un módulo de dicha aplicación, dicho módulo será luego importado dentro del fichero **views.py** de la aplicación de tele-visita y será donde en función de las necesidades se crearán las sesiones (salas), previamente se comprobará que el usuario es un tutor docente, se almacenará en la base de datos para no tener que volver a crearla, posteriormente se crearan los tokens (llaves) para los usuarios autorizados puedan acceder a la sala en cuestión.
Dicha API se descarga desde la web de Opentok, en mi proyecto está incluido con el nombre de [OpentokSDK.py](#) dentro de la aplicación tele-visita.
- La otra API es del lado del cliente y se encarga de publicar o conectar/desconectarse a los diferentes streams de audio y video. Disponible en Objective-C, JavaScript, y ActionScript 3.0 . Yo he elegido la API [JavaScript](#).

2.2.3.- Arquitectura de Opentok



*JavaScript developers use the OpenTok JavaScript library.
Flash developers use the OpenTok ActionScript library.

Recientemente Opentok incluye de modo aún experimental una segunda API del lado del cliente, dicha API utiliza WebRTC (Web Real-Time Communication). WebRTC está siendo elaborada por la World Wide Web Consortium (W3C) para permitir a las aplicaciones del navegador realizar llamadas de voz, chat de vídeo y uso compartido de archivos P2P sin plugins. La API que pone a disposición Opentok y que utiliza esta tecnología sólo se puede utilizar todavía si disponemos del navegador Chrome 23 o superior y limita cada sala a la presencia de hasta 4 participantes.

2.2.4.- [WebRTC](#)

Parece ser el futuro en cuanto a las aplicaciones de comunicaciones en tiempo real ejecutadas dentro del navegador web, ya que proporciona mejor experiencia de usuario, mayor calidad de audio/video, necesitando para ello menor ancho de banda.

He incluido la posibilidad de utilizar este modo de videoconferencia, dentro de cada sala de tutor docente, para ello habrá que pulsar sobre el icono de WebRTC que aparece dentro de cada sala, posteriormente se podrá volver al modo normal pulsando sobre el icono de Flash.

Dentro de cada sala, incluida la Sala común o Sala de espera, se dispone también de una Pizarra-Chat, en la que los participantes que se encuentren en esa sala podrán escribir conjuntamente un documento en tiempo real y/o chatear, sobre todo por si hubiese problemas de audio-video en la videoconferencia.

Para llevar a cabo la Pizarra-Chat he utilizado [Etherpad-Lite](#), un editor colaborativo en tiempo real generado a partir de Etherpad, pero mucho más ligero y más estable que el Etherpad original., ya que incluye una base de código más pequeño, manejable y bien documentado.

Trabaja conjuntamente con Node.js, por lo que deberá estar previamente instalado en el servidor, como luego explicaré.

Una vez instalado la creación, interacción y modificación de cada Pizarra-Chat (en la jerga de Etherpad son PADS), la hago a través de un [plugin jquery](#) que estará embebido dentro de la plantilla Django correspondiente.

Todos los detalles sobre la instalación y administración del proyecto los describiré en el punto número 4 del presente documento, en el apartado sobre "Manual de Instalación y administración."

3.- FASE DE PRUEBAS

Para poner a prueba la aplicación he creado los 4 roles de usuarios que pueden interaccionar con la aplicación:

- Administrador
- Tutor docente
- Tutor laboral
- 2 o más alumnos

Utilizando los diferentes usuarios y creando registros desde los diferentes roles he podido probar desde qué usuarios fallaban o no los diferentes elementos que conforman la aplicación. Pero sin duda lo que mas me ha ayudado para ver y solucionar los diferentes problemas ha sido ponerla a prueba en el periodo de FCT del curso 2012-2013 tanto en la primera como tercera evaluación.

En la primera evaluación, es cierto que aún faltaban bastantes cosas para terminarla, pero el uso más básico si se podía ya utilizar, es decir:

- Para el tutor docente: El registro de las empresas y alumnos, creación de contratos y programas formativos. Posibilidad de tele-visitar al alumno y/o tutor laboral. Ver anotaciones diarias de los alumnos en su centro de trabajo.
- Para el alumno: Anotar diariamente las actividades realizadas en su centro de trabajo, poder generar PDF semanalmente.

Durante este periodo pudieron hacerse varias pruebas sobre todo de la aplicación de tele-visita, pudiéndose comprobar, como era de esperar, que la videoconferencia necesita una conexión a Internet estable, que no tenga excesivo retardo, etc,...para que la comunicación sea fluida.

En concreto pudieron realizarse 3 pruebas con el alumno Manuel Limón, que se encontraba realizando su periodo de FCT en la empresa **Agua y Estructuras S.A. (Ayesa)** en Sevilla, en 2 de dichas ocasiones, se utilizó la conexión a Internet del Instituto, una vez en horario de tarde, en la que se pudo realizar medianamente bien, y en la otra ocasión en horario de mañana, en la que la calidad de la videoconferencia era insufrible, teniendo finalmente que utilizar un móvil con conexión a Internet.

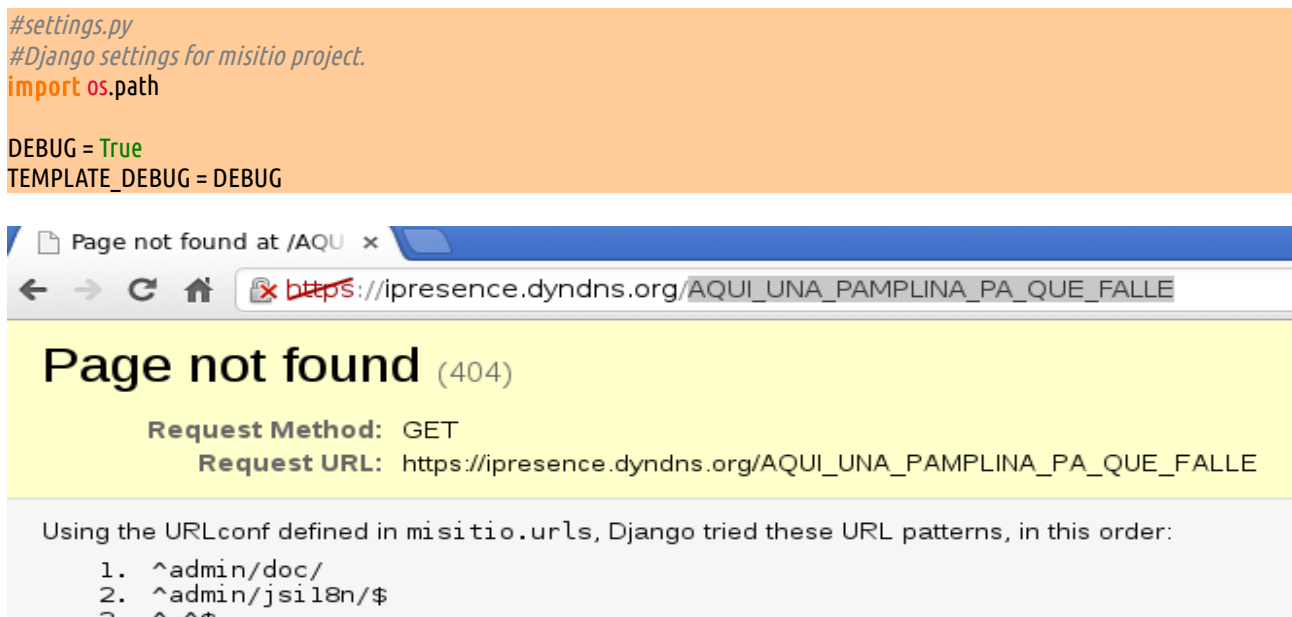
El motivo de la baja calidad de la conexión a Internet es un problema si se pretende realizar tele-visitas desde allí, ya que en la mayoría de ocasiones falla y no se logra realizar una videoconferencia decente, en muchos casos ni siquiera establecer una primera conexión. Esto es debido a dos motivos:

- Una vez que salimos de la infraestructura de red del departamento ICO, al parecer, por capturas de tráfico realizadas, debe existir un Hub/concentrador de red, el cual sería muy conveniente localizarlo y sustituirlo por un switch.
- Mala calidad, en cuanto velocidad, retardo, etc,..de la propia conexión a Internet del centro.

De todas formas lo que influye en mayor medida es el primer punto, ya que fastidia directamente la videoconferencia, eliminando la sensación de tiempo real por completo. Se hizo una tercera prueba desde fuera de las instalaciones del Instituto, utilizándose conexiones ADSL y la mejora fue considerable. En esta ocasión intervinieron además 3 participantes y no 2 como anteriormente.

Ha sido durante el periodo de la tercera evaluación, cuando se ha podido probar completamente la aplicación sobre todo en el apartado de documentación relacionado con la FCT, para ello han intervenido todos los alumnos de 2º de ASIR que se encontraban en el periodo de FCT: Adri, Manu, Sara, Ana, Yasmina, Emilio, Alberto, Farru, Rocio y los profesores: Jaime, Diego y Javier.

Durante todo este periodo la aplicación ha funcionado en modo DEBUG, siendo así posible ver por donde fallaba la aplicación.



Para ponerla en producción simplemente habrá que modificar el valor a **False** y crear las correspondientes plantillas para errores 404 y 500.

4.- DOCUMENTACION DEL SISTEMA

4.1.- Introducción a la aplicación.

La aplicación ha sido diseñada para centralizar y recoger la información referente a la FCT de los alumnos de 2º de ASIR del IES Fernando Aguilar, agilizar la realización de todo ese trabajo y poder tele-visitarse a los alumnos y/o tutores laborales.

Existen cuatro grupos diferenciados los cuales podrán hacer un uso determinado de la aplicación: el administrador, los tutores docentes, tutores laborales y los alumnos.

- El administrador será quien instalará y configurará la aplicación, elegirá el servicio de alojamiento mas apropiado para la aplicación, instalará, configurará y mantendrá todo lo necesario para que funcione la aplicación (Servidor Web, SGBD,...), creará los primeros usuarios de la aplicación: los tutores docentes.
- Los tutores docentes: Serán los encargados de dar de alta a los alumnos, las empresas colaboradoras así como sus diferentes centros de trabajo, gerentes, tutores laborales asociados y finalmente será el encargado de recabar la información acerca de los contratos de colaboración, crear los programas formativos.
Asimismo podrá como anteriormente he comentado tele-visitarse a los alumnos y/o tutores laborales, supervisar las anotaciones diarias de sus alumnos.
Puede darse el caso que alguno/s de los tutores docentes también sea administrador.
- Los tutores laborales: Tan sólo podrán supervisar que las anotaciones que sus alumnos realizan en la plataforma se adecuan a la labor diaria en el centro de trabajo.
- Los alumnos: Su uso se limita a ir anotando sus quehaceres diarios en su centro de trabajo.

A continuación voy a pasar a describir los diferentes manuales de usuario (para los tutores docentes/laborales y los alumnos) y los manuales de instalación y administración (para el administrador/es)

4.2.- Manual de instalación.

Será el administrador el encargado de realizar esta labor, deberá empezar por la instalación de los paquetes necesarios:

```
sudo apt-get install python-django apache2 php5 libapache2-mod-php5 postgresql python-psycopg2 phpPgadmin  
libapache2-mod-wsgi python-setuptools python-reportlab python-html5lib python-pypdf gzip git-core curl python libssl-dev  
pkg-config build-essential abiword sqlite3 python-pip python-imaging git
```

```
sudo easy_install pisa  
sudo a2enmod rewrite ssl proxy proxy_http  
sudo a2ensite default-ssl
```

Instalar NodeJS (no instalar desde repositorios oficiales porque da problemas) para ello añadir el siguiente PPA:

```
sudo apt-get update
sudo apt-get install python-software-properties python g++ make
sudo add-apt-repository ppa:chris-lea/node.js
sudo apt-get update
sudo apt-get install nodejs
```

Este repositorio además de instalar NODEJS instalara también NPM.

Descomprimir la última versión proporcionada del proyecto ipresence.tar.gz en el directorio personal del usuario administrador (SUDOER):

```
tar -xvzf ipresence.tar.gz
```

Hay que asegurarse que se encuentre dentro de **misitio** el fichero **wsgi_handler.py** el cual debe contener:

```
ubuntu@ubuntu:~$ cat /home/ubuntu/misitio/wsgi_handler.py
```

```
import os, sys
sys.path.append(os.path.dirname(os.path.abspath(__file__)) + '/../')
os.environ['DJANGO_SETTINGS_MODULE'] = 'misitio.settings'
import django.core.handlers.wsgi
application = django.core.handlers.wsgi.WSGIHandler()
```

Antes de continuar, vamos a instalar y configurar Etherpad-lite, para ello en primer lugar vamos a crear un usuario llamado etherpad-lite:

```
ubuntu@ubuntu:~$ sudo adduser etherpad-lite
[sudo] password for ubuntu:
Añadiendo el usuario `etherpad-lite' ...
Añadiendo el nuevo grupo `etherpad-lite' (1001) ...
Añadiendo el nuevo usuario `etherpad-lite' (1001) con grupo `etherpad-lite' ...
Creando el directorio personal `/home/etherpad-lite' ...
Copiando los ficheros desde `/etc/skel' ...
Introduzca la nueva contraseña de UNIX:
Vuelva a escribir la nueva contraseña de UNIX:
passwd: password updated successfully
Changing the user information for etherpad-lite
Enter the new value, or press ENTER for the default
    Full Name []:
    Room Number []:
    Work Phone []:
    Home Phone []:
    Other []:
¿Es correcta la información? [S/n] S
```

Nos cambiamos a dicho usuario:

```
ubuntu@ubuntu:~$ su etherpad-lite
```

Contraseña:

```
etherpad-lite@ubuntu:/home/ubuntu$ cd
```

```
etherpad-lite@ubuntu:~$
```

Nos situamos sobre el directorio que queremos instalar Etherpad-lite y clonamos con Git el repositorio de dicho proyecto:

```
etherpad-lite@ubuntu:~$ git clone 'git://github.com/ether/etherpad-lite'
```

Cloning into 'etherpad-lite'...

remote: Counting objects: 17883, **done**.

remote: Compressing objects: 100% (5438/5438), **done**.

remote: Total 17883 (delta 12895), reused 16957 (delta 12115)

Receiving objects: 100% (17883/17883), 4.58 MiB | 1.29 MiB/s, **done**.

Resolving deltas: 100% (12895/12895), **done**.

Nos situamos sobre el directorio del proyecto descargado:

```
etherpad-lite@ubuntu:~$ cd etherpad-lite/
```

Y ejecutamos el comando `/home/etherpad-lite/etherpad-lite/bin/run.sh` para configurarlo y actualizar las dependencias:

```
etherpad-lite@ubuntu:~$ /home/etherpad-lite/etherpad-lite/bin/run.sh
```

Tras un rato quedará todo configurado, este comando alertará sobre posibles problemas.

El anterior comando hará que se cree el fichero de configuración en: `/home/etherpad-lite/etherpad-lite/settings.json`, el cual necesitaremos editar para adaptar Etherpad-lite a nuestro sistema, SGBD, etc,...

Pararlo con `Ctrl + C` y seguir configurando, podemos ejecutar el siguiente comando si queremos utilizar PostgreSQL como resguardo de las sesiones y no DirtyDB como por defecto trae:

```
etherpad-lite@ubuntu:~/etherpad-lite$ npm install pg
```

El anterior comando de instalación suele dar unas advertencias y/o errores, los ignoramos.

Creamos, con el usuario `sudoer` una base de datos llamada **store** en PostgreSQL para Etherpad-lite, le damos todos los permisos para el usuario de Postgre **toni** y reiniciamos PostgreSQL:

```
ubuntu@ubuntu:~/ sudo -u postgres createdb store
```

```
ubuntu@ubuntu:~$ sudo -u postgres psql
```

```
psql (9.1.9)
```

```
Type "help" for help.
```

```
postgres=# grant all privileges on database store to toni;
```

```
GRANT
```

```
postgres=# \q
```



```
ubuntu@ubuntu:~$ sudo service postgresql restart
```

```
* Restarting PostgreSQL 9.1 database server
```

Y editamos el fichero de configuración de Etherpad-lite con los datos de nuestra nueva base de datos, por defecto Etherpad-lite usa DirtyDB, luego hay que comentar las líneas que hacen referencia a él:

```
ubuntu@ubuntu:~$ sudo nano /home/etherpad-lite/etherpad-lite/settings.json
```

```
//"dbType" : "dirty",
//the database specific settings
//"dbSettings" : {
//      "filename" : "var/dirty.db"
//      },

// PostgreSQL Configuration
"dbType" : "postgres",
"dbSettings" : {
    "user" : "toni",
    "host" : "localhost",
    "password": "tucontraseña",
    "database": "store"
},
```

Y ejecutamos de nuevo el comando anterior para observar posibles problemas:
/home/etherpad-lite/etherpad-lite/bin/run.sh

```
etherpad-lite@ubuntu:~$ /home/etherpad-lite/etherpad-lite/bin/run.sh
```

A continuación nos cambiamos de nuevo al usuario sudoer, en nuestro caso el usuario ubuntu y creamos un directorio para los registros del servicio etherpad:

```
etherpad-lite@ubuntu:~$ su ubuntu
```

Contraseña:

```
ubuntu@ubuntu:/home/etherpad-lite$ cd
```

```
ubuntu@ubuntu:~$ sudo mkdir /var/log/etherpad-lite
```

Nos aseguramos que el usuario etherpad-lite tiene acceso a dicho directorio:

```
ubuntu@ubuntu:~$ sudo chown etherpad-lite:etherpad-lite /var/log/etherpad-lite
```

Seguidamente debemos configurar etherpad-lite para que funcione como un servicio, para ello, dentro del directorio **/etc/init.d/** creamos el script **etherpad-lite**:

```
ubuntu@ubuntu:~$ sudo nano /etc/init.d/etherpad-lite
```

...el cual debe contener:

```
#!/bin/sh
### BEGIN INIT INFO
# Provides:      etherpad-lite
# Required-Start: $local_fs $remote_fs $network $syslog
# Required-Stop: $local_fs $remote_fs $network $syslog
```

```

# Default-Start: 2 3 4 5
# Default-Stop: 0 1 6
# Short-Description: starts etherpad lite
# Description: starts etherpad lite using start-stop-daemon
### END INIT INFO
PATH="/usr/local/sbin:/usr/local/bin:/sbin:/bin:/usr/sbin:/usr/bin:/opt/node/bin"
LOGFILE="/var/log/etherpad-lite/etherpad-lite.log"
EPLITE_DIR="/home/etherpad-lite/etherpad-lite"
EPLITE_BIN="bin/safeRun.sh"
USER="etherpad-lite"
GROUP="etherpad-lite"
DESC="Etherpad Lite"
NAME="etherpad-lite"
set -e
. /lib/lsb/init-functions
start() {
    echo "Starting $DESC..."
    start-stop-daemon --start --chuid "$USER:$GROUP" --background --make-pidfile --pidfile /var/run/$NAME.pid --exec
$EPLITE_DIR/$EPLITE_BIN -- $LOGFILE$
    echo "done"
}
#We need this function to ensure the whole process tree will be killed
killtree() {
    local _pid=$1
    local _sig=${2-TERM}
    for _child in $(ps -o pid --no-headers --ppid $_pid); do
        killtree $_child $_sig
    done
    kill -$_sig $_pid
}
stop() {
    echo "Stopping $DESC..."
    while test -d /proc/$(cat /var/run/$NAME.pid); do
        killtree $(cat /var/run/$NAME.pid) 15
        sleep 0.5
    done
    rm /var/run/$NAME.pid
    echo "done"
}
status() {

```

```

status_of_proc -p /var/run/$NAME.pid "" "etherpad-lite" && exit 0 || exit $?
}
case "$1" in
  start)
    start
    ;;
  stop)
    stop
    ;;
  restart)
    stop
    start
    ;;
  status)
    status
    ;;
  *)
    echo "Usage: $NAME {start|stop|restart|status}" >&2
    exit 1
    ;;
esac
exit 0

```

Le damos permisos de ejecución:

```
ubuntu@ubuntu:~$ sudo chmod +x /etc/init.d/etherpad-lite
```

Habilitarlo como servicio con el update-rc.d:

```

ubuntu@ubuntu:~$ cd /etc/init.d/
ubuntu@ubuntu:/etc/init.d$ sudo update-rc.d etherpad-lite defaults
Adding system startup for /etc/init.d/etherpad-lite ...
/etc/rc0.d/K20etherpad-lite -> ../init.d/etherpad-lite
/etc/rc1.d/K20etherpad-lite -> ../init.d/etherpad-lite
/etc/rc6.d/K20etherpad-lite -> ../init.d/etherpad-lite
/etc/rc2.d/S20etherpad-lite -> ../init.d/etherpad-lite
/etc/rc3.d/S20etherpad-lite -> ../init.d/etherpad-lite
/etc/rc4.d/S20etherpad-lite -> ../init.d/etherpad-lite
/etc/rc5.d/S20etherpad-lite -> ../init.d/etherpad-lite

```

De esta forma etherpad-lite se ejecutará al iniciarse, de todas formas para arrancarlo y pararlo, simplemente:

```
ubuntu@ubuntu:~$ sudo /etc/init.d/etherpad-lite start
```

Starting Etherpad Lite...

done

```
ubuntu@ubuntu:~$ sudo /etc/init.d/etherpad-lite stop
```

Stopping Etherpad Lite...

done

A continuación muestro cómo deo los ficheros de configuración de los sitios Apache:

```
ubuntu@ubuntu:~$ sudo nano /etc/apache2/sites-available/default
```

```
<VirtualHost *:80>
    RewriteEngine On
    RewriteCond %{SERVER_PORT} !^443$
    RewriteCond %{REQUEST_URI} !^/p/
    RewriteCond %{REQUEST_URI} !^/static/
    RewriteCond %{REQUEST_URI} !^/pluginfw/
    RewriteCond %{REQUEST_URI} !^/javascripts/
    RewriteCond %{REQUEST_URI} !^/socket.io/
    RewriteCond %{REQUEST_URI} !^/ep/
    RewriteCond %{REQUEST_URI} !^/minified/
    RewriteCond %{REQUEST_URI} !^/api/
    RewriteCond %{REQUEST_URI} !^/ro/
    RewriteCond %{REQUEST_URI} !^/error/
    RewriteCond %{REQUEST_URI} !^/jserror
    RewriteCond %{REQUEST_URI} !/favicon.ico
    RewriteCond %{REQUEST_URI} !/robots.txt
    RewriteRule ^/(.*) https://%{SERVER_NAME}/$1 [L,R]
    ServerAdmin you@example.com
    ServerName example.com

    # Django settings
    WSGIScriptAlias / /home/ubuntu/misitio/wsgi_handler.py
    WSGIDaemonProcess misitio user=ubuntu group=ubuntu processes=1 threads=10

    <IfModule mod_proxy.c>
        ProxyVia On
        ProxyRequests Off
        ProxyPass / http://localhost:9001/
        ProxyPassReverse / http://localhost:9001/
        ProxyPreserveHost on
    </Proxy *>

    Options FollowSymLinks MultiViews
```

```

        AllowOverride All
        Order allow,deny
        allow from all
    </Proxy>
</IfModule>
</VirtualHost>

```

También el siguiente fichero:

```
ubuntu@ubuntu:~$ sudo nano /etc/apache2/sites-available/default-ssl
```

```

<IfModule mod_ssl.c>
    <VirtualHost _default_:443>
        ServerAdmin you@example.com
        ServerName example.com
        DocumentRoot /home/ubuntu/misitio
        # Django settings
        WSGIScriptAlias / /home/ubuntu/misitio/wsgi_handler.py
        WSGIProcessGroup misitio
        # Non-Django directories
        Alias /static /home/ubuntu/misitio/static/
        <Location "/static">
            SetHandler None
        </Location>
        ErrorLog ${APACHE_LOG_DIR}/error.log
        LogLevel warn
        CustomLog ${APACHE_LOG_DIR}/ssl_access.log combined
        SSLEngine on
        SSLCertificateFile /etc/ssl/certs/ipresence.dyndns.org.crt
        SSLCertificateKeyFile /etc/ssl/private/ipresence.dyndns.org.key

        <FilesMatch "\.(cgi|shtml|phtml|php)$">
            SSLOptions +StdEnvVars
        </FilesMatch>
        <Directory /usr/lib/cgi-bin>
            SSLOptions +StdEnvVars
        </Directory>
        BrowserMatch "MSIE [2-6]" \
            nokeepalive ssl-unclean-shutdown \
            downgrade-1.0 force-response-1.0
        BrowserMatch "MSIE [17-9]" ssl-unclean-shutdown
    </VirtualHost>
</IfModule>

```

Para crear el certificado auto-firmado al que hace referencia en el fichero de configuración anterior:

```
ubuntu@ubuntu:~$ sudo openssl req -new -x509 -nodes -days 3650 -out /etc/ssl/certs/ipresence.dyndns.org.crt -keyout /etc/ssl/private/ipresence.dyndns.org.key
```

Generating a 1024 bit RSA private key

.....++++++

.....++++++

writing new private key to '/etc/ssl/private/ipresence.dyndns.org.key'

You are about to be asked to enter information that will be incorporated into your certificate request.

What you are about to enter is what is called a Distinguished Name or a DN.

There are quite a few fields but you can leave some blank

For some fields there will be a default value,

If you enter '.', the field will be left blank.

Country Name (2 letter code) [AU]:ES

State or Province Name (full name) [Some-State]:Spain

Locality Name (eg, city) []:Cadiz

Organization Name (eg, company) [Internet Widgits Pty Ltd]:ipresence.dyndns.org

Organizational Unit Name (eg, section) []:La aplicacion web de telepresencia de FCT

Common Name (e.g. server FQDN or YOUR name) []:ipresence.dyndns.org

Email Address []:

```
ubuntu@ubuntu:~$ sudo service apache2 restart
```

Restarting web server apache2 [OK]

Configurar PostgreSQL:

Editar /etc/postgresql/9.1/main/postgresql.conf

```
ubuntu@ubuntu:~$ sudo nano /etc/postgresql/9.1/main/postgresql.conf
```

dejando...

```
#listen_addresses = 'localhost'
```

así:

```
listen_addresses = '*'
```

...descomentar la linea para dejar:

```
password_encryption = on
```

Luego establecer contraseña usuario postgres:

```
ubuntu@ubuntu:~$ sudo -u postgres psql template1
```

```
psql (9.1.9)
```

```
Type "help" for help.
```

```
template1=# ALTER USER postgres with encrypted password 'tucontraseña';
```

```
ALTER ROLE
```

```
template1=# \q
```

Crear usuario **toni** y base de datos **presence** y darle privilegios a toni sobre esta:

```
ubuntu@ubuntu:~$ sudo -u postgres createuser
```

```
Enter name of role to add: toni
```

```
Shall the new role be a superuser? (y/n) n
```

```
Shall the new role be allowed to create databases? (y/n) n
```

```
Shall the new role be allowed to create more new roles? (y/n) n
```

```
ubuntu@ubuntu:~$ sudo -u postgres createdb presence
```

```
ubuntu@ubuntu:~$ sudo -u postgres psql
```

```
psql (9.1.9)
```

```
Type "help" for help.
```

```
postgres=# alter user toni with encrypted password 'tucontraseña';
```

```
ALTER ROLE
```

```
postgres=# grant all privileges on database presence to toni;
```

```
GRANT
```

```
postgres=# \q
```

Para que los usuarios usen autenticación MD5 editar:

```
ubuntu@ubuntu:~$ sudo nano /etc/postgresql/9.1/main/pg_hba.conf
```

```
# Database administrative login by Unix domain socket
```

```
local          all             postgres        peer
```

```
local          presence        toni            md5
```

Reiniciar:

```
ubuntu@ubuntu:~$ sudo service postgresql restart
```

```
* Restarting PostgreSQL 9.1 database server
```

[OK]

Configurar PHPPGADMIN: Por defecto solo se puede acceder desde localhost, para que deje acceder desde otras ubicaciones editar el siguiente fichero de configuración: **/etc/phppgadmin/apache.conf**

```
ubuntu@ubuntu:~/misitio$ sudo nano /etc/phppgadmin/apache.conf
```

y dejar así:

```
#allow from 127.0.0.0/255.0.0.0 ::1/128
```

```
allow from all
```

Y por último creamos las tablas del modelo Django:

```
ubuntu@ubuntu:~/misitio$ python manage.py syncdb
```

Creating tables ...

Creating table auth_permission

Creating table auth_group_permissions

Creating table auth_group

Creating table auth_user_user_permissions

Creating table auth_user_groups

Creating table auth_user

Creating table auth_message

Creating table django_content_type

Creating table django_session

Creating table presence_empresa

Creating table presence_gerente

Creating table presence_centro

Creating table presence_laboral

Creating table presence_docente

Creating table presence_alumno

Creating table presence_curso

Creating table presence_aprendizaje

Creating table presence_contratoprograma_aprendizajes

Creating table presence_contratoprograma

Creating table presence_seguimiento

Creating table presence_visita_aprendizajesPositivos

Creating table presence_visita

Creating table presence_usuarioinactivo

Creating table televisita_room

Creating table django_admin_log

You just installed Django's auth system, which means you don't have any superusers defined.

Would you like to create one now? (yes/no): yes

Username (Leave blank to use 'ubuntu'): toni

E-mail address: toni@toni.com

Password:

Password (again):

Superuser created successfully.

Installing custom SQL ...

Installing indexes ...

Installed 6 object(s) from 1 fixture(s)

Tras todo esto, la aplicación y todos sus componentes deben estar funcionando. Es muy importante revisar el fichero **settings.py** del proyecto, pues ahí se definen algunas constantes importantes para que todo funcione bien:

- Modificar SERVER por la IP pública, url o servicio dinámico DNS asociado, si la aplicación se usase en un entorno LAN, sustituir por la IP del servidor.

```
SERVER = 'ipresence.dyndns.org'
```

- Modificar DATABASES por los parámetros de configuración de su SGBD.

```
DATABASES = {  
    'default': {  
        'ENGINE': 'django.db.backends.postgresql_psycopg2',  
        'NAME': 'presence',  
        'USER': 'toni',  
        'PASSWORD': 'tucontraseña',  
        'HOST': '',  
        'PORT': '',  
    }  
}
```

- Modificar los parámetros de configuración de su servidor de correo saliente. Yo he utilizado una cuenta de correo de Gmail.

```
DEFAULT_FROM_EMAIL = 'i-Presence <ipresence.cadiz@gmail.com>'  
EMAIL_HOST = 'smtp.gmail.com'  
EMAIL_HOST_USER = 'ipresence.cadiz@gmail.com'  
EMAIL_HOST_PASSWORD = 'tucontraseña'  
EMAIL_PORT = 587  
EMAIL_USE_TLS = True
```

4.3.- Manual de administración.

Tras la instalación de la aplicación, al administrador le queda ya poco que hacer, simplemente dar de alta a los tutores docentes, los cuales continuarán con la labor de dar de alta a las empresas, a los alumnos, tutores laborales, etc,...

Para dar de alta a los tutores docentes, tras autenticarse con la cuenta administrador creada en el proceso de instalación de la aplicación (tras ejecutar el comando **python manage.py syncdb**), será dirigido directamente al panel de administración de la aplicación (...no al panel de administración de Django).

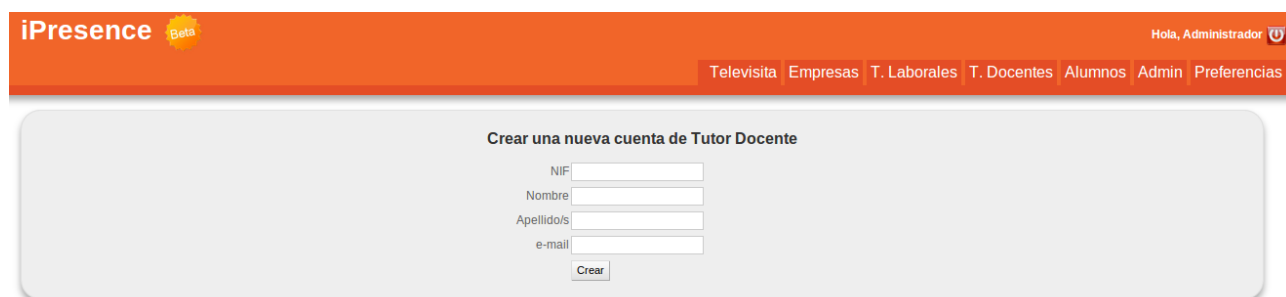


Luego pulsar sobre el enlace Tutores Docentes y finalmente darlo de alta.



Nombre	Apellidos	e-mail	Último Login	Activo	Videochat
Fco. Javier	García Estevez	javier.gestevez@iesfernandoagular.es	04/Jun/2013 19:34	✓	✓
TUTOR DOCENTE	DE PRUEBA	docente.ipresence@gmail.com	12/Jun/2013 20:58	✓	✓
Jaime	Canillas Galiano	jaime.canillas@iesfernandoagular.es	06/Jun/2013 08:10	✓	✓
Diego	Vegas Pérez	diego.vegas@yahoo.es	10/Jun/2013 12:42	✓	✓

Rellenar los datos del tutor y listo.



Crear una nueva cuenta de Tutor Docente

NIF

Nombre

Apellido/s

e-mail

Resaltar que el administrador tendrá también acceso a panel de administración de Django, para ello deberá pulsar sobre el enlace **Admin**.

Administración de DjangoBienvenido/a, **Administrador**.

Sitio administrativo

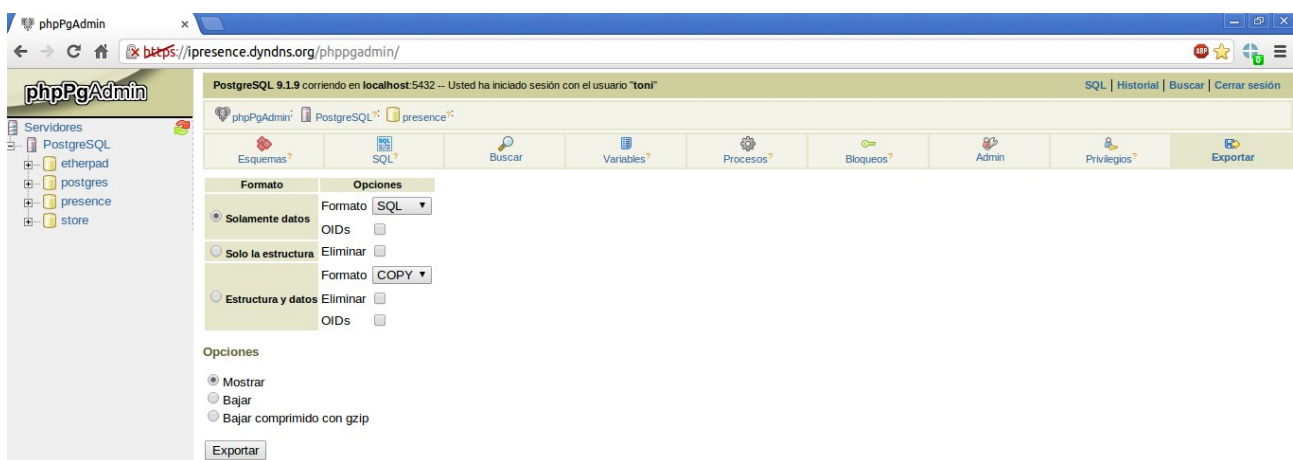
Auth	
Grupos	+ Añadir ✎ Modificar
Usuarios	+ Añadir ✎ Modificar

Presence	
Alumnos	+ Añadir ✎ Modificar
Aprendizajes	+ Añadir ✎ Modificar
Centros	+ Añadir ✎ Modificar
Contrato programas	+ Añadir ✎ Modificar
Cursos	+ Añadir ✎ Modificar
Docentes	+ Añadir ✎ Modificar
Empresas	+ Añadir ✎ Modificar
Gerentes	+ Añadir ✎ Modificar
Laborals	+ Añadir ✎ Modificar
Seguimientos	+ Añadir ✎ Modificar
Visitas	+ Añadir ✎ Modificar

Acciones recientes
Mis acciones

- ✖ 2013/2014
Curso
- ✖ 2013/2014
Curso
- ✎ Curso: 2012/2013 Alumno:
Alumno de Prueba Z Empresa:
Zalt - EMPRESA DE PRUEBA, SL
Contrato programa
- ✖ Alumno de Prueba 2 Z
Alumno
- ✖ 2013/2014
Curso
- ✎ Alumno de Prueba 2 Z
Alumno
- ✎ Alumno de Prueba Z
Alumno
- ✎ ALUMNO2 ZZ
Alumno
- ✎ Fco. Javier García Estevez
Docente
- ✖ alumno kllkjl
Alumno

Otra de las cosas de las que deberá estar pendiente el administrador es hacer las convenientes copias de seguridad de la base de datos, para ello podrá utilizar la herramienta que mas le agrade, desde herramientas de administración gráfica como pgAdmin III, phpPgAdmin, PGAccess, entre [otras](#), o utilizando las herramientas que proporciona PostgreSQL (**pg_dump**), yo he utilizado **phpPgAdmin**.



4.4.- Manual de usuario.

Voy a dividir el manual de usuario en cuatro apartados, por un lado el manual de usuario según los tres roles de la aplicación (alumno, T. docente y T. laboral) y a continuación el manual de usuario de lo concerniente a la videoconferencia que es igual para todos los roles.

En primer lugar, y es común para todos los usuarios, cada usuario recibirá un e-mail, con una invitación a activar su cuenta de usuario, realizada o bien por el administrador de la aplicación o por un tutor docente, el procedimiento ya está descrito en el punto 2.1.9. Será por tanto el punto de entrada y la primera actuación de cada usuario en la aplicación.

4.4.1.- Manual del tutor docente

Cada tutor docente, tras autenticarse, será dirigido por defecto al apartado donde se recogen todas las anotaciones diarias de sus alumnos en la FCT.























Podrá dar de alta a un nuevo alumno pinchando sobre el enlace “Alumnos” y luego sobre “Dar de alta a nuevo Alumno”, donde rellenará un formulario con los datos de éste.

Será también el docente el encargado de dar de alta a las empresas, para ello deberá pulsar sobre el enlace “Empresas” y luego sobre “Nueva Empresa” donde rellenará un formulario con los datos de ésta.

Por defecto se creará un centro de trabajo con los mismos datos de la empresa que se acaba de introducir, pero si necesitamos asociar mas centros de trabajo asociados a esta empresa debemos sobre el listado de empresas pulsar sobre el enlace “Nuevo Centro de Trabajo” de la empresa en cuestión.

Nueva Empresa

Buscar:























EMPRESAS		Centros		Gerentes		Tutores		Contratos		Alumnos
	 AFT Informática, S.L.									
	 Agua y Estructuras, S.A.									

También será necesario añadir los gerentes y los tutores laborales asociados a cada empresa, pulsando simplemente en cada icono “+” de la columna en cuestión. El otro icono de cada columna es para listar el total de Centros, Gerentes o tutores de cada empresa.

Una vez que se tengan los datos previos necesarios para crear un contrato-programa_formativo, que son, empresa, centro de trabajo, gerente, tutores y alumno, es hora de dar de alta un contrato-programa asociada a una empresa en cuestión, para ello pulsar sobre el icono “+” de la columna contratos.

Nueva Empresa

Buscar:

EMPRESAS		Centros		Gerentes		Tutores		Contratos		Alumnos
	 AFT Informática, S.L.									
	 Agua y Estructuras, S.A.									

Tras pinchar sobre este icono la aplicación primero comprobará que en la pestaña cursos existe el curso actual, y si no es así lo crea:

```
from misitio.presence.models import Curso
from time import strftime
@login_required
def compruebaCurso(request):
    mesactual = strftime("%m")
    yearActual = int(strftime("%Y"))
    if mesactual in ('01','02','03','04','05','06'):
        try:
            curso = Curso.objects.get(fecha_inicio=yearActual-1)
        except:
            Curso.objects.create(fecha_inicio=yearActual-1)
    elif mesactual in ('09','10','11','12'):
        try:
            curso = Curso.objects.get(fecha_inicio=yearActual)
        except:
            Curso.objects.create(fecha_inicio=yearActual)
```

Para crear un contrato-programa asociado a un alumno habrá que buscarlo en el desplegable de alumnos, donde sólo aparecerán aquellos alumnos (activos) que aún no tienen contrato-programa asociado en el curso actual.

Es recomendable que aquellos usuarios que han cursado y aprobado el periodo de FCT se marquen como usuario no activo en la aplicación y de esta forma no aparecerá más en este desplegable.

The screenshot shows the iPresence application interface. At the top, there is a navigation bar with the iPresence logo and a user profile section for 'Hola, TUTOR DOCENTE'. Below the navigation bar, there is a main content area with a title 'Registrar nuevo Contrato con Zalt - EMPRESA DE PRUEBA, SL'. The form contains several dropdown menus and checkboxes. The 'Curso' dropdown is set to '2012/2013'. The 'Tutor docente' dropdown is set to 'TUTOR DOCENTE DE PRUEBA'. The 'Alumno' dropdown is empty. The 'Tutor laboral' dropdown is empty. The 'Gerente' dropdown is empty. The 'Modalidad' dropdown is empty. The 'Periodo' dropdown is empty. Below these dropdowns, there are five checkboxes for 'Resultados de aprendizaje esperados':

- ☐ Identifica la estructura y organización de la empresa relacionándolas con el tipo de servicio que presta.
- ☐ Aplica hábitos dicos y laborales en el desarrollo de su actividad profesional de acuerdo con las características del puesto de trabajo y con los procedimientos establecidos en la empresa.
- ☐ Organiza el procedimiento de trabajo que debe desarrollar, interpretando la documentación específica.
- ☐ Determina las características técnicas de la instalación a partir de las funcionalidades y necesidades establecidas.
- ☐ Participa en el diseño, la puesta en marcha y el mantenimiento de instalaciones con servicios de red local e Internet, documentando la intervención realizada.
- ☐ Asiste a los usuarios resolviendo problemas de la explotación del sistema, según las normas y tiempos establecidos.

Below the checkboxes, there are two text input fields labeled 'Actividades Formativo-Productivas' and 'Criterios de evaluación'. At the bottom right of the form, there is a 'Crear' button.

Habr  que seleccionar al tutor laboral y al gerente, donde s lo aparecer n los asociados a esa empresa. Elegir tambi n la modalidad y periodo de FCT adecuados.

Los tres  ltimos campos corresponden con el programa formativo asociado al contrato formativo, sencillamente habr  que indicar lo que proceda en cada uno de los campos.

Una vez creado el contrato-programa,  ste podr  ser editado, para ello habr  que pulsar sobre el listado total de contratos asociados a cada empresa y pulsar sobre el enlace "Editar" del contrato-programa en cuesti n.

iPresence <small>Beta</small>							
Hola, TUTOR DOCENTE 							
Televisita Alumnos Empresas Fichas Preferencias							
Relaci�n de contratos con Universidad de C�diz							
Curso	Alumno	Visitas		Programa/Contrato			
2012/2013	Antonio Ruiz Rond�n						
2012/2013	Roc�o Rubio Conchas						
2012/2013	Miguel A. Farruja Porras						

Cada tutor docente s lo podr  editar aquellos contrato-programa que  l haya creado, de lo contrario le aparecer  el mensaje:

!! Error !!

No eres el actual Tutor de este alumno

Para posteriormente ver y/o descargar tanto los datos del contrato como los del programa formativo habr  que pulsar sobre los enlaces marcados a continuaci n.

iPresence <small>Beta</small>							
Hola, TUTOR DOCENTE 							
Televisita Alumnos Empresas Fichas Preferencias							
Relaci�n de contratos con Universidad de C�diz							
Curso	Alumno	Visitas		Programa/Contrato			
2012/2013	Antonio Ruiz Rond�n						
2012/2013	Roc�o Rubio Conchas						
2012/2013	Miguel A. Farruja Porras						

O bien también se pueden descargar los PDF pulsando sobre el icono indicado dentro del panel de visualización de cada contrato-programa.

A.- Programa Formativo

JUNTA DE ANDALUCÍA

CONSEJERÍA DE EDUCACIÓN

FORMACIÓN EN CENTROS DE TRABAJO. PROGRAMA FORMATIVO

(Hoja de)

CENTRO DOCENTE: IES Fernando Aguilar Quignon		CENTRO DE TRABAJO COLABORADOR: Universidad de Cádiz	
PROFESORÍA RESPONSABLE DEL SEGUIMIENTO: Fco. Javier García Estevez		TUTOR/A DEL CENTRO DE TRABAJO: Luis Nadal de Mora	
ALUMNO/ALUMNA: Miguel A. Farrujia Porras		PERÍODO DE REALIZACIÓN DE LA FCT	
		PRIMER TRIMESTRE	SEGUNDO TRIMESTRE X TERCER TRIMESTRE
CURSO ESCOLAR: 2012/2013			
FAMILIA PROFESIONAL: Informática		CICLO FORMATIVO: Adm. de Sist. Inf. en Red	
		GRADO: Superior	
RESULTADOS DE APRENDIZAJE	ACTIVIDADES FORMATIVO-PRODUCTIVAS	CRITERIOS DE EVALUACIÓN	
Identifica la estructura y organización de la empresa relacionándolas con el tipo de servicio que presta. Aplica hábitos éticos y laborales en el desarrollo de su actividad profesional de acuerdo con las características del puesto de trabajo y con los procedimientos establecidos en la empresa. Organiza el procedimiento de trabajo que debe desarrollar, interpretando la documentación específica. Determina las características técnicas de la instalación a partir de las funcionalidades y necesidades establecidas. Participa en el diseño, la puesta en marcha y el mantenimiento de instalaciones con servicios de red local e internet, documentando la intervención realizada. Asiste a los usuarios resolviendo problemas de la explotación del sistema, según las normas y tiempos establecidos.	Estudio de la infraestructura de red de la UCA, así como participación en su mantenimiento y resolución de incidencias. Soporte a usuarios. Instalación y configuración de sistemas para aulas y talleres	Los incluidos en el apartado 5.1 de la PD del módulo	

En Cádiz a 8 de abril de 2013

B.- Datos del Contrato

DATOS DE LA EMPRESA

CIF: B11557618 RAZON SOCIAL: Zalt - EMPRESA DE PRUEBA, SL
 DIRECCION: Rto. Interior Zona Franca
 POBLACION: Cádiz C. POSTAL: 11007
 TELEFONO: 956897845 FAX: 956784512 E-MAIL: tedeum@tedeum.com
 GERENTE D. Sebas Jimenez Sánchez DNI: 31258741Q
 TUTOR LABORAL D. TUTOR LABORAL DE PRUEBA DNI: 31258742V
 TUTOR DOCENTE D. TUTOR DOCENTE DE PRUEBA DNI: 123456S

CENTRO DE TRABAJO

DIRECCION: Rto. Interior Zona Franca
 POBLACION: Cádiz C. POSTAL: 11007
 TELEFONO: 956897845 FAX: 956784512 E-MAIL: tedeum@tedeum.com

DATOS ALUMNO

APELLIDOS Y NOMBRE: Z, Alumno de Prueba
 DNI: 47123456K TELEFONO: 956789456 MOVIL: 656656656
 FECHA DE NACIMIENTO: 1 de enero de 1977 MODALIDAD DE AYUDA ¹: B

- ¹ A. Misma localidad desde 5 Kms
 B. Otra localidad hasta 20 Kms
 C. Otra localidad a más de 20 Kms





Una vez realizado todo lo anterior sólo queda ir observando las anotaciones realizadas por sus alumnos, para ello pulsar sobre el enlace "Fichas".

4.4.2.- Manual del alumno

El uso de la aplicación por parte del alumno se limita en exclusiva, a poder usar la videoconferencia e ir anotando sus actividades diarias, para ello tras autenticarse será dirigido directamente a la sección "Fichas" y verá un listado de todas las actividades diarias anotadas o bien podrá insertar una nueva entrada para un día concreto pulsando sobre "Anotar nueva tarea".

Anotar nueva tarea

CENTRO DOCENTE: IES Fernando Aguilar Quignon TUTOR DOCENTE: Fco. Javier García Estevez	CENTRO PRACTICAS: Universidad de Cádiz TUTOR LABORAL: Luis Nadal de Mora
ALUMNO/A: Antonio Ruiz Rondán	CICLO FORMATIVO: Administración de Sistemas Informáticos en Red GRADO: S

Fecha	Actividad	Tiempo	Observaciones	Acciones
viernes 07/Jun/2013	Instalación y configuración de software de telefonía IP (SIP) al personal docente de la Escuela Superior de Ingeniería.	6 horas		
jueves 06/Jun/2013	Instalación y configuración de software de telefonía IP (SIP) al personal docente de la Escuela Superior de Ingeniería.	6 horas		
miércoles 05/Jun/2013	Instalación y configuración de software de telefonía IP (SIP) al personal docente de la Escuela Superior de Ingeniería.	6 horas		
martes 04/Jun/2013	Instalación y configuración de software de telefonía IP (SIP) al personal docente de la Facultad de Filosofía y Letras	6 horas		

Para ver el listado de anotaciones de una semana concreta habrá que seleccionar dentro del formulario de búsqueda cualquier día de la semana en cuestión y ya la aplicación se encargará de extraer los días comprendidos de lunes a viernes de esa semana, si la semana está completa, dará la opción de generar un PDF de esa semana.

Listado de Tareas Alumno: Antonio Ruiz Rondán

Fecha 06/05/2013

Hoy






Mostrar

Generar PDF

Tareas de la semana del lunes 06-mayo-2013 al viernes 10-mayo-2013

Anotar nueva tarea

CENTRO DOCENTE: IES Fernando Aguilar Quignon TUTOR DOCENTE: Fco. Javier García Estevez	CENTRO PRACTICAS: Universidad de Cádiz TUTOR LABORAL: Luis Nadal de Mora
ALUMNO/A: Antonio Ruiz Rondán	CICLO FORMATIVO: Administración de Sistemas Informáticos en Red GRADO: S

Fecha	Actividad	Tiempo	Observaciones	Acciones
viernes 10/May/2013	Instalación y configuración de software de telefonía IP (SIP) al personal docente de la Facultad de Empresariales. Modificación de la estructura del RAC de la Fac. de Filosofía y Letras, eliminando los transformadores.	6 horas		
jueves 09/May/2013	Día de huelga	0 horas		
miércoles 08/May/2013	Instalación y configuración de software de telefonía IP (SIP) al personal docente de la Facultad de Económicas.	8 horas		
martes 07/May/2013	Instalación y configuración de software de telefonía IP (SIP) al personal docente de la Facultad de Económicas.	6 horas		
lunes 06/May/2013	Instalación y configuración de software de telefonía IP (SIP) al personal docente de la Facultad de Económicas.	6 horas		

Para ver de nuevo todas las anotaciones habrá que darle de nuevo al buscador sin seleccionar una fecha concreta.

El PDF que se genera tiene el formato: **FichaSemanal-Nombre Ape1 Ape2-fecha1 al fecha2.pdf**

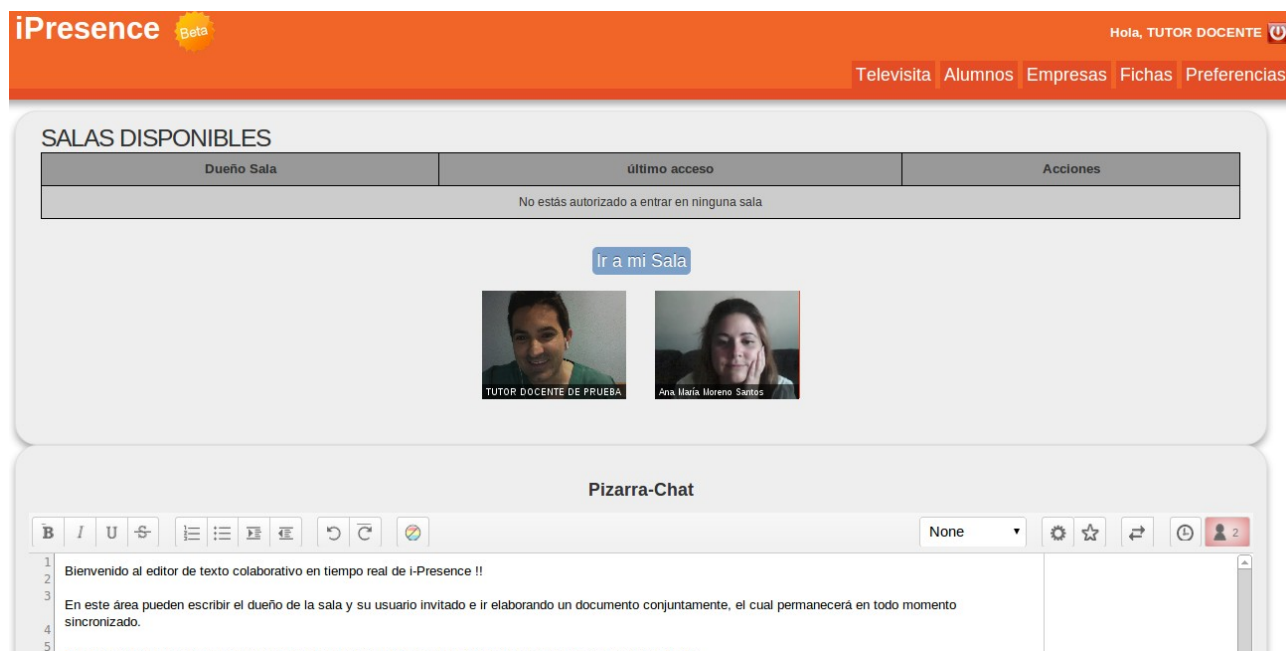
4.4.3.- Manual del tutor laboral

El uso que el tutor laboral puede hacer de la aplicación es para poder video-conferenciarse con el tutor docente de cada alumno o para ver las anotaciones diarias de las actividades realizadas por sus alumnos, del mismo modo que lo hace el tutor docente.

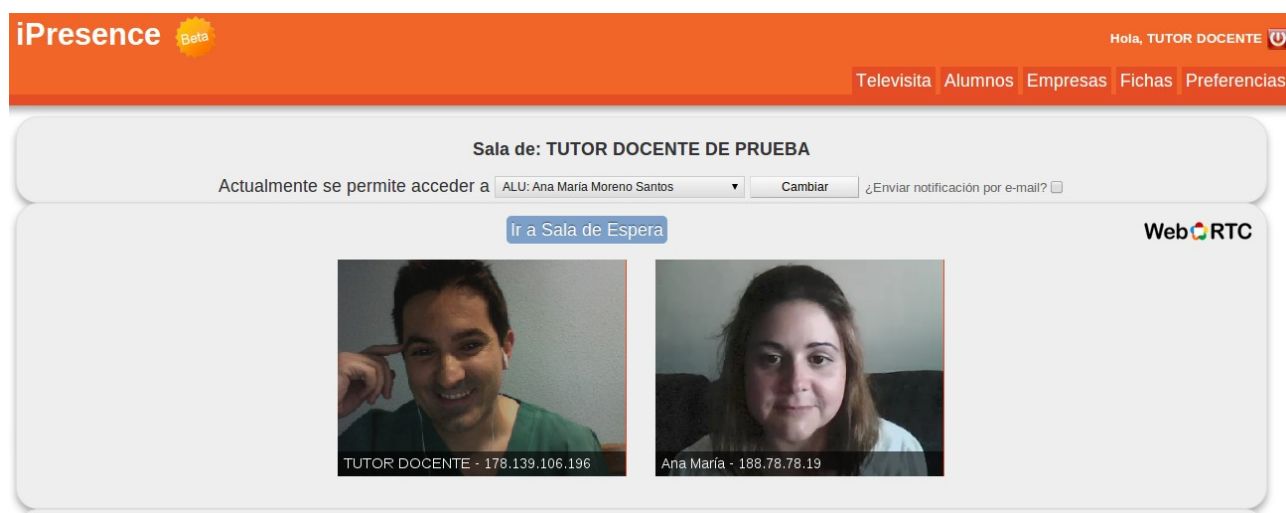
4.4.4.- Videoconferencia

Una vez autenticados, para hacer una videoconferencia hay que pinchar en el enlace **TELEVISITA**, y accederás en un primer momento a una "Sala de Espera" donde aparecerá una tabla indicándote las Salas a las que te está permitido acceder.

O bien pincha directamente a "Ir a mi Sala" eres tutor docente.



Cada tutor docente tiene su propia sala y es el tutor docente el que invita a cada alumno o a cada tutor laboral a acceder a su propia sala, por tanto es posible que no estés invitado a acceder a ninguna sala.



Mientras tanto puedes permanecer en la "Sala de espera".

Los tutores docentes también pueden ser invitados a la sala de otro tutor docente.

La aplicación en principio debe de reconocer la webcam y micrófono y aparecer un dialogo como el siguiente:



En cuyo caso debes presionar sobre Allow, para permitir que tu navegador web, acceda, gracias al Plug-in de adobe flash, a tu cámara y micrófono.

Puede ocurrir que ese cuadro dialogo no te aparezca, o que tengas cualquier problema para realizar la videoconferencia, por tanto tienes que asegurarte primero de las siguientes aspectos:

- Comprueba con [Test_1](#) que tienes el Plug-in de Adobe Flash Player 10.3 o superior (recomendado 11+). Si no lo tienes [descárgatelo de aqui](#) e instálalo.

Si usas Ubuntu o similar, lo mejor es hacerlo desde los repositorios con:

```
sudo apt-get install flashplugin-installer
```

- Asegúrate que tu navegador sea Internet Explorer 8 / Firefox 3 / Chrome 13 o superior. [Test_2](#)
- Debes tener abiertos los siguientes puertos de salida: 80, 443 y 1935, comprueba que indica Connection established en cada uno de ellos al pulsar sobre [Test_3](#)

Network ↳ Port Access

Required	Your setup
Port 1935 should be open	Connection established
Port 80 should be open	Connection established
Port 443 should be open	Connection established

- Necesitas también poder acceder a las URLs que aparecen al pulsar el [Test_4](#) y que en todas indica Connected Successfully

Network

↳ URI Access

Required	Your setup
Allow access to "http://static.opentok.com"	Connected Successfully
Allow access to "http://api.opentok.com/hl"	Connected Successfully
Allow access to "http://hlg.tokbox.com"	Connected Successfully

- Lógicamente debes tener correctamente instalados y configurados la webcam y el micrófono. Pula en el botón [Test_5](#) y comprueba que ambos son detectados correctamente




Hardware

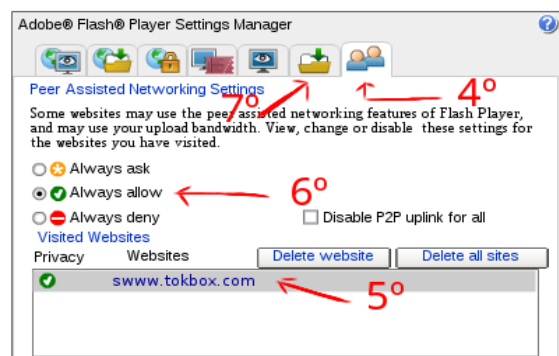
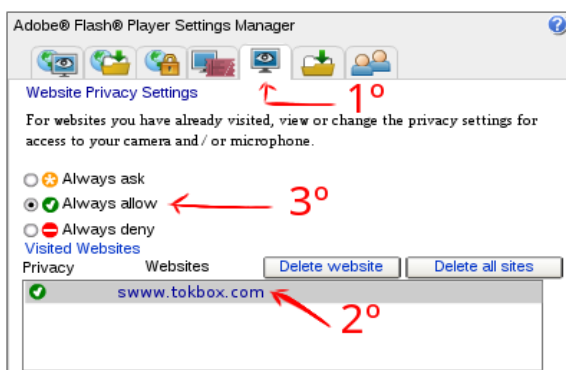
Camera, microphone



Contents

Camera Count	✓ Camera detected	More info
Microphone Count	✓ Microphone detected	More info

- Si tu webcam y micrófono son detectados, pero aún así sigue sin aparecerte en i-Presence, sigue los 7 pasos que aparecen a continuación tras pulsar el botón [Test_6](#) y así la aplicación (el navegador) podrá acceder a tu webcam y micrófono correctamente.
 - Paso 1º: Pincha en la pestaña  donde dice **Website Privacy Settings**
 - Paso 2º: Busca la url **swww.tokbox.com** y márcala
 - Paso 3º: Marca donde dice **Always allow**
 - Paso 4º: Pincha en la última pestaña  la que dice **Peer Assisted Networking Settings**
 - Paso 5º: Busca también la url **swww.tokbox.com** y márcala
 - Paso 6º: Señala también **Always allow**
 - Paso 7º: Finalmente pincha en la  penúltima pestaña para que se guarden todos los cambios.



5.- CONCLUSIONES FINALES

El principal objetivo de este proyecto era modelar los datos que se manejan en el módulo de FCT del ciclo de ASIR del centro, así como establecer los cimientos para una correcta gestión del mismo.

Particularmente mi principal motivación desde el principio fue profundizar en el uso del framework Django que estudiamos en el módulo de Implantación de Aplicaciones Web (IAW).

Por otro lado, dotar al nuevo sistema de una administración centralizada de la información referente a la FCT, construyendo una aplicación que posibilitara este cometido.

Realizar una mejor gestión de los datos, así como reducir y centralizar el trabajo, así como de la labor realizada con los alumnos del módulo de FCT.

Se ha realizado un producto capaz de cumplir con la administración planteada en los inicios del proyecto, contando siempre con herramientas no privativas, lo que no suponen costo adicional alguno al disponer de licencias Open-Source, pero que pueden ofrecer un alto rendimiento en los productos realizados.

El proyecto está abierto a futuras ampliaciones y mejoras, para lo cual pongo a disposición del Instituto el código fuente, backup de Postgresql (a 13-junio-2013) y ficheros de configuración utilizados.

6.- BIBLIOGRAFIA

- [La guía definitiva de Django](#) Holovaty, Adrian; Kaplan-Moss, Jacob, (aut.) Carreira Carreira, Oliver, (tr.) de Anaya Multimedia.
- DjangoProject: <https://www.djangoproject.com>
- Documentación Python: <http://docs.python.org/release/2.7.3>
- Opentok: <http://www.tokbox.com/opentok>
- Documentación PostgreSQL: <http://www.postgresql.org>
- Ubuntu: <https://help.ubuntu.com/12.04/serverguide/index.html>
- Etherpad-lite: <https://github.com/ether/etherpad-lite>
- y sobre todo... Stack Overflow: <http://stackoverflow.com>