Universidad Nacional de Río Cuarto
Facultad de Ciencias Exactas,Fco-Qcas y Natulares
Departamento de Computación
Proyecto(Código 1998)

Proyecto para optar al título de analista en computación

Proyecto : Minuto gol

Versión : 0.0.1

Integrantes : Gustavo Martínez, DNI: 33833434

Índice

Registro de cambios:

Fecha	Autor	Versión	Comentarios	Horas
23/06/2015	Gustavo Martinez	0.0.1	Inicio de documento	6
1/09/2015	Gustavo Martinez	0.0.1	Documentacion de requerimientos	8
17/09/2015	Gustavo Martinez	0.0.1	Documentación del testing	7

Registro de aprobación:

Firma	Responsable	Título	Fecha

1-Introducción

En este capítulo se darán algunos conceptos importantes para la comprensión de los temas involucrados en el desarrollo del trabajo. Se presenta una descripción del propósito que lleva desarrollar el trabajo, el alcance del mismo acompañado de la bibliografia consultada y del equipo de trabajo que paticipa en el mismo.

1.1 - Propósito

El presente documento tiene como propósito definir las especificaciones funcionales, no funcionales y del sistema para la implementación de una aplicación web que permitirá solicitar turnos correspondientes a canchas en complejos deportivos, como también permitirá administrar complejos deportivos.

Los inconvenientes que surgen a la hora de realizar reservaciones, es que actualmente las unicas dos formas de llevar acabo dicha actividad es por medio de una comunicación telefónica que la misma implica costo de comunicación o personalmente obligando a un cliente de un complejo a presentarse fisicamente para solicitar un turno a una cancha de un determinado complejo. Otros de los inconvenientes que surgen es la administración de complejos deportivos que implican llevar un registro de torneos, canchas, fixtures entre otros manualmente a través de cuadernos que dificultan acceso y control a lo largo del tiempo.

Es por estas razones que es necesario poder digitalizar las actividades mencionadas para poder proveer una opción mucho más comoda para todos los clientes de los complejos deportivos como así tambien poder failitar la tarea de administración a los encargados de los diferentes complejos deportivos

1.2 – Alcance

MG será una aplicación web basada en DJANGO que funcionará sobre una arquitectura WEB que permitirá gestionar la administración de complejos deportivos y la solicitud de turnos a canchas entre usuarios de la plataforma, la cual además permite una conexión de multi-usuarios.

Ésta aplicación dará apoyo a los siguientes procesos:

- Gestionar cuentas de usuarios(ABM)
- Gestionar complejos(ABMLB)
- Gestionar equipos(ABMLB)
- Gestionar torneos(ABMLB)
- Gestionar canchas(ABMLB)
- Gestionar reservas(ABML)
- Gestionar publicidades(ABML)
- Gestionar estadisticas de cada jugador por torneo a los que participa(ABMLB)

- Gestionar ciudades (ABM)
- Gestionar fixtures (ABMLB)
- Gestionar partidos (ABMLB)
- Buscar fixtures/estadisticas/complejos/equipos/torneos/partidos
- Enviar e-mail's a la administración de MG

A través de MG, NO se permitirá:

- Un usuario UCM no podrá realizar ABM de complejos.
- Un usuario UPR no podrá modificar o eliminar complejos ajenos.
- Un usuario UPR no podrá realizar ABM de equipos.
- Un usuario UCM no podrá modificar o eliminar equipos en los cuales no es el capitán de dichos equipos.
- Un usuario UCM no podrá realizar ABM de torneos.
- Un usuario UPR no podrá modificar o eliminar torneos correspondientes a complejos deportivos en los cuales no es el propietario.
- Un usuario UCM no podrá notificar la asistencia de una reserva realizada como así tampoco podrá cancelar las reservaciones correspondientes a otros usuarios

- Un usuario UCM no podrá realizar ABM de publicidades.
- Un usuario UPR no podrá realizar modificar o eliminar publicidades que no sean de su propiedad.
- Un usuario UCM no podrá realizar ABM de estadisticas.
- Un usuario UPR no podrá modificar o eliminar estadisticas de jugadores a torneos correpondientes a complejos en los cuales no es el propietario.
- Un usuario UCM no podrá realizar ABM de fixtures
- Un usuario UPR no podrá modificar fixtures correspondientes a torneos en los cuales no es el propietario del complejo en el cual se llevará a cabo.
- Un usuario UCM no podrá realizar ABM de canchas
- Un usuario UPR no podrá realizar ABM de canchas a complejos en los cuales no es el propietario.
- Un usuario no podrá modificar ni eliminar una cuenta de usuario en la cual no sea el propietario.

1.3 - Definiciones, acrónimos y abreviaciones

- MG : minuto gol
- UCM : usuario común
- UPR : usuario propietario
- ABM : alta-baja-modificación

- ABMLB : alta-baja-modificación-listar-buscar
- ABML : alta-baja-modificación-listar
- CRUD : Create-Read-Update-Delete(crear-leer-actualizar-eliminar)
- Multi-usuarios: En general se le llama multiusuario a la característica de un sistema operativo o programa que permite proveer servicio y procesamiento a múltiples usuarios simultáneamente.
- Gestionar: Todo lo relacionado con la gestión significa realizar Altas, Bajas, Modificaciones y consultas de alguna información necesaria por el usuario vía internet .
- Usuario: Persona que puede ingresar a Minuto Gol como
    - UCM
    - UPR
    - usuario administrador
    - usuario invitado
- Base de Datos: Es un conjunto de datos que pertenecen al mismo contexto almacenados sistemáticamente para su posterior uso.
- Django: Es un framework basado en el lenguaje Python que permite agilizar el proceso de producción de código para proyectos de mediano y gran tamaño.
- Python: Es un lenguaje de programación orientado a objetos. Que permite el desarrollo de una aplicación para algún uso específico.
- Internet: Es un método de interconexión descentralizada de redes de computadoras implementado en un conjunto de protocolos denominado TCP/IP.
- Boostrap : es un framework HTML,CSS y JavaScript que podemos utilizar como base para crear aplicaciones web.
- startUml : es un software que permite realizar diagramas uml.
- jquery : es una biblioteca de JavaScript, que permite simplificar la manera de interactuar con los documentos HTML, manejar eventos, desarrollar animaciones y agregar interacción a páginas web.

- Datepicker : es un conjunto de funcionalidades de jquery para el modelado de calendarios dinámicos

- Planner : Herramienta que permite desarrollar diagramas de gantt, la cual está disponible para la distibución Linux.

1.4 – Bibliografía

- IEEE 830 esp - http://www.ctr.unican.es/asignaturas/is1/IEEE830_esp.pdf
- http://www.getbootstrap.com/ (Framework html-css-javascript) ultimo acceso : 1/09/2015
- Libro Python para todos por Raúl González Duque
- El libro de Django por Adrian Holovaty y Jacob Kaplan-Moss
- https://www.djangoproject.com/ versión 1.6.8 último acceso: 1/09/2015
- https://docs.djangoproject.com/en/1.6/topics/testing/ útimo acceso 17/09/2015
- https://jqueryui.com/datepicker/ último acceso 16/09/2015

1.5– Personal involucrado

Nombre	Gustavo Martínez
Rol	Team
Categoría profesional	Developer
Responsibilidades	Desarrollado y testing

Nombre	Marcela Daniele
Rol	Manager
Categoría profesional	Lic. en Sistemas
Responsibilidades	Toma de decisiones

Nombre	Marcelo Uva
Rol	Manager
Categoría profesional	Lic. en Sistemas
Responsibilidades	Toma de decisiones

Visión global : En el presente documento se encontrará la información acerca de las características del producto de software, interfaces del usuario, interfaces del sistema, características de los usuarios, descripción de los requerimientos funcionales, no funcionales y del sistema, los cuales se representarán mediante el siguiente formato:

Id	Indentificador de un requerimiento específico
Nombre	Título del requerimiento
Descripción	Descripción del requrimiento
Estimación	Estimación del tiempo necesario para implementar un requerimiento
Valor para el cliente	Imporatncia del requerimiento para el cliente
Condiciones de satisfacción	Descripción de las actividades que realiza el requerimiento
Rendimiento	-Restricciones de diseño e implementación -Interfaces Externas

2-Descripción general

En este capítulo se describirá la perspectiva del producto como así también las funcionalidades provistas, tipos de usuarios con la definición de los permisos de cada usuario

2.1- Perspectiva del producto

El Sistema MG será un producto que trabajará sobre la arquitectura **web**, además trabajará de forma independiente con lo que no interactuará con otros sistemas.

2.2- Funciones del producto

El sistema MG permitirá realizar las siguientes funciones:

- Creación, actualización y eliminación de cuentas de usuarios.
- Creación, actualización, eliminación, busqueda y consulta de complejos.
- Creación, actualización, eliminación, busqueda y consulta de canchas.
- Creación, actualización, eliminación, busqueda y consulta de partidos.
- Creación, actualización, eliminación, busqueda y consulta de torneos.
- Creación, actualización, eliminación, consulta de reservas.
- Creación, actualización, eliminación, busqueda y consulta de estadisticas.
- Creación, actualización, eliminación, busqueda y consulta de fixture.
- Creación, actualización, eliminación, consulta de publicidades.
- Solicitud de turnos a canchas perteneciente a complejos deportivos.
- Creación, actualización, eliminación, busqueda y consulta de equipos.
- Envío de e-mails a la administración de sistema.

2.3-Características del usuario

El sistema MG contendrá 4 tipos de usuarios que interactúan y lo administran: Usuario invitado, Usuario común, Usuario propietario y el usuario administrador del sistema.

2.3.1-Perfil de usuario

Cada usuario tendrá un perfil específico para que su interacción con el sistema sea correcta y no conlleve a fallos:

- Usuario invitado: Persona que puede utilizar el sistema con permisos limitados. Puede solo realizar determinadas búsquedas y visitar complejos como así también enviar emails a la admistración del sistema, pero sin los permisos para poder realizar ningún tipo de modificaciones sobre el estado de sistema.

- Usuario común: Persona que tiene permisos para realizar solicitudes de turnos a canchas, administración de equipos, consultar estadística de cada torneo a los cuales participa,consultar torneos y fixtures.

- Usuario propietario: Persona que tiene los permisos para administrar complejos deportivos, canchas, partidos, fixtures, torneos y publicidades. Como así también controlar las reservaciones correspondientes a las canchas de sus complejos deportivos y gestionar las reservaciones a un determinado usuario.

- usuario administrador : Usuario con gran conocimiento en el manejo del sistema. Encargado de activar cuentas de usuario-propietario, alta de ciudades, controlar las publicidades, complejos, canchas, torneos, fixtures, equipos, reservaciones, partidos y cumplir la función de administrador de la aplicación en su totalidad.

2.3.2- Jerarquía de usuarios

> Usuario administrador

Usuario administrador

>Usuario propietario

Usuario propietario

>Usuario común

Usuario común

>Usuario invitado

Usuario invitado

2.4-Restricciones generales

2.4.1 -Políticas Reguladoras

La aplicación se desarrollara mediante software de licencia libre por lo tanto no se deberá pagar por el uso de: servidor WEB (APACHE), Sistema de Gestión de Base de Datos (MySQL), el lenguaje de programación (PYTHON), el framework (DJANGO) ,el framework (BOOSTRAPT) por lo tanto, la utilización de estos programas se hará mediante las políticas establecidas por este tipo de licenciamiento.

2.4.2 -Limitaciones de Hardware

Para esta aplicación será necesario un computador servidor en el cual se instalara el servidor WEB Apache, MySQL, PYTHON , DJANGO,BOOSTRAPT y la aplicación MG.

2.4.3-Interfaces con otras aplicaciones

Debido a que el sistema no interactúa con otros sistemas y es autónomo no se desarrollan interfaces con otras aplicaciones. Las conexiones necesarias para la utilización del servidor web, MySQL, python, django, boostrapt y un DNS, se hará por medio de la configuración de estos programas.

2.4.4- Funcionamiento Paralelo

El sistema MG no cuenta con esta característica.

2.4.5 -Funciones de Auditoria

El software así como el proceso será auditado por la directora Marcela Daniele y el co-director Marcelo Uva, ambos managers del proyecto.

2.4.6 -Funciones de Control

El sistema debe controlar los permisos que tiene cada usuario para su accesibilidad de una manera correcta, de tal forma que pueda acceder a la información que le corresponde de acuerdo a su rol.

Debe tener controles adecuados para la validación de datos.

2.4.7 -Requisitos del Lenguaje

Todo el material que se realice para el usuario y la aplicación debe de estar en lenguaje Español.

2.4.8 -Protocolos Señalados

Se usará protocolos de comunicación TCP/IP.

2.4.9-Credibilidad de la aplicación

Para garantizar una buena credibilidad el sistema deberá ser sometido a una serie de pruebas para establecer que se encuentra acorde a los requerimientos que se plasman en el documento en tanto a la consistencia de datos como al rendimiento de la aplicación, tales como tiempos de respuesta razonables.
Se utilizará el módulo de testing que trae incorporado django para realizar las pruebas de unidades a cada modulo que constituye el sistema.

2.4.10 -Consideraciones de seguridad

Cada usuario deberá autenticarse y su acceso verificado por el sistema. Todas las claves de seguridad deberán estar seguras y en su defecto encriptadas en la base de datos para dar una buena seguridad al sistema y su información.

3-Requerimientos específicos

En este capítulo se describirá los requerimientos de interfaces externas, requerimientos funcionales de la aplicación como también los requerimientos no funcionales (como seguridad, fiabilidad, disponibilidad, mantenibilidad y portabilidad).

3.1-Requerimientos de interfaces externas

El sistema MG no tendrá interconexión con otros sistemas de información, por lo tanto no es necesaria la utilización de interfaz alguna.
La relación con un servidor Web, DNS y Gestor de Base de Datos se hará a través de los archivos de configuración de estos.

3.2 Requerimientos funcionales

3.2.1-Ingreso a la aplicación

Id	1
Nombre	Como usuario invitado quiero registrarme como usuario común
Descripción	Se permitirá a los Usuarios invitado gestionar una cuenta de usuario-común.
Estimación	
Valor para el cliente	
Condiciones de Satisfacción	Al finalizar la operación, los cambios solicitados se ven reflejados en el sistema.
Rendimiento	**Restricciones de Diseño e Implementación:** -La alta de un usuario-común por parte de un usuario invitado se permitirá unicamente en caso de que no exista un usuario-común con dichos datos.

Id	2
Nombre	Como usuario invitado quiero registrarme como usuario propietario
Descripción	Se permitirá a los Usuarios invitado gestionar una cuenta de usuario-propietario.
Estimación	
Valor para el cliente	
Condiciones de Satisfacción	Al finalizar la operación, los cambios solicitados se ven reflejados en el sistema.
Rendimiento	**Restricciones de Diseño e Implementación:** -La alta de un usuario-propietario por parte de un usuario invitado se permitirá unicamente en caso de que no exista un usuario-común con dichos datos.La activación de la cuenta será llevada a cabo por la administración de MG

Id	3
Nombre	Como un usuario registrado quiero modificar mi perfil de usuario
Descripción	Se permitirá a los Usuarios registrados actualizar la información de su perfil.
Estimación	
Valor para el cliente	
Condiciones de Satisfacción	Al finalizar la operación, los cambios solicitados se ven reflejados en el sistema.
Rendimiento	**Restricciones de Diseño e Implementación:** -La modificación de los datos de perfil se permitira unicamente si el usuario no es un usuario invitado y si es el propietario de la cuenta que intenta modificar.

Id	4
Nombre	Como un usuario registrado quiero poder darme de baja
Descripción	Se permitirá a los usuarios registrados poder eliminar su cuenta
Estimación	
Valor para el cliente	
Condiciones de Satisfacción	Al finalizar la operación, los cambios solicitados se ven reflejados en el sistema.
Rendimiento	**Restricciones de Diseño e Implementación:** -Para la baja de un usuario se deberá estar logueado, además podra eliminar unicamente su propia cuenta, es decir un usuario no podrá eliminar una cuenta que no sea de su propiedad.

Id	5
Nombre	Como usuario propietario quiero poder CRUD publicidades
Descripción	Se permitirá a los Usuarios propietarios gestionar publicidades relacionada a sus complejos.
Estimación	
Valor para el cliente	
Condiciones de Satisfacción	Al finalizar la operación, los cambios solicitados se ven reflejados en el sistema.
Rendimiento	**Restricciones de Diseño e Implementación:** -las publicidades relacionadas con temáticas no relacionadas con el deportes y los complejos de un usuario seran eliminadas por parte de la admisnistracion de MG

Id	6
Nombre	Como usuario logueado quiero ver publicidades de diferentes complejos
Descripción	Se permitirá a los usuarios logueados enterarse de diferentes noticias de los complejos,como asi tambien torneos entre otros temas a partir de publicidades.
Estimación	
Valor para el cliente	
Condiciones de Satisfacción	Al finalizar la operación, los cambios solicitados se ven reflejados en el sistema.
Rendimiento	**Restricciones de Diseño e Implementación:** -La visualización de las distintas publicidades debe realizar de manera aleatoria aproximadamente, brindando la misma posibilidad a todas las publicidades de visualizarse.

Id	7
Nombre	Como usuario quiero poder ver complejos deportivos
Descripción	Se permitirá a todo usuario del MG ver un listado de complejos disponibles.
Estimación	
Valor para el cliente	
Condiciones de Satisfacción	Al finalizar la operación, los cambios solicitados se ven reflejados en el sistema.
Rendimiento	

Id	8
Nombre	Como usuario propietario quiero poder registrar complejos deportivos
Descripción	Se permitirá a los Usuarios propietario gestionar complejos deportivos.
Estimación	
Valor para el cliente	
Condiciones de Satisfacción	Al finalizar la operación, los cambios solicitados se ven reflejados en el sistema.
Rendimiento	**Restricciones de Diseño e Implementación:** -La alta de un complejo por parte de un usuario propietario se permitirá unicamente en caso de que, dicho usuario sea parte del staff(es decir tiene permisos suficientes para la creación de complejos).

Id	9
Nombre	Como usuario propietario quiero poder actualizar complejos deportivos.
Descripción	Se permitirá a los Usuarios propietario actualizar la información de complejos deportivos.
Estimación	
Valor para el cliente	
Condiciones de Satisfacción	Al finalizar la operación, los cambios solicitados se ven reflejados en el sistema.
Rendimiento	**Restricciones de Diseño e Implementación:** -La actualizacion de complejos deportivos se permitirá, unicamente si el usuario tiene permisos de propietario y ademas es el propietario del complejo

Id	10
Nombre	Como usuario propietario quiero poder eliminar complejos deportivos
Descripción	Se permitira realizar la eliminación de complejos deportivos
Estimación	
Valor para el cliente	
Condiciones de Satisfacción	Al finalizar la operación, los cambios solicitados se ven reflejados en el sistema.
Rendimiento	**Restricciones de Diseño e Implementación:** -La baja de un complejo deportivo se podrá llevar a cabo unicamente si el usuario tiene los permisos de propietario y ademas es el propietario del complejo a eliminar.

Id	11
Nombre	Como usuario quiero poder buscar complejos
Descripción	Se permitirá a un usuario poder bucar un complejo
Estimación	
Valor para el cliente	
Condiciones de Satisfacción	Al finalizar la operación, los cambios solicitados se ven reflejados en el sistema.
Rendimiento	**Restricciones de Diseño e Implementación:** -L a busqueda de un complejo se podrá llevar a cabo a través de un determinado patrón de busqueda.

Id	12
Nombre	Como usuario administrador quiero poder CRUD ciudades
Descripción	Se permitirá gestionar ciudades.
Estimación	
Valor para el cliente	
Condiciones de Satisfacción	Al finalizar la operación, los cambios solicitados se ven reflejados en el sistema.
Rendimiento	**Restricciones de Diseño e Implementación:** -La alta/baja/actualizacion/lectura podrá ser realizada unicamente por aquellos usuarios con permisos de administrador. -Todo usuario con permisos de usuario común o propietario solo podra disponer de dicha información unicamente para indicar su lugar de procedencia a la hora de registrarse a MG.

Id	13
Nombre	Como usuario quiero poder comunicarme vía email con la administración de MG
Descripción	Se permitirá a los usuarios enviar un email a la administración
Estimación	
Valor para el cliente	
Condiciones de Satisfacción	Al finalizar la operación, los cambios solicitados se ven reflejados en el sistema.
Rendimiento	**Restricciones de Diseño e Implementación:** -El envío de un email debe contener una dirección de correo electrónico válida.

Id	14
Nombre	Como usuario quiero poder listar canchas de diferentes complejos
Descripción	Se permitirá a los usuarios logueados visualizar información de canchas.
Estimación	
Valor para el cliente	
Condiciones de Satisfacción	Al finalizar la operación, los cambios solicitados se ven reflejados en el sistema.
Rendimiento	**Restricciones de Diseño e Implementación:** -Para poder acceder a la visualización de diferentes canchas, el usuario logueado debe ser un usuario propietario o un usuario común.

Id	15
Nombre	Como usuario propietario quiero poder CUD canchas.
Descripción	Se permitirá a los usuarios propietarios registrar,actualizar y borrar canchas correspondientes a un determinado complejo.
Estimación	
Valor para el cliente	
Condiciones de Satisfacción	Al finalizar la operación, los cambios solicitados se ven reflejados en el sistema.
Rendimiento	**Restricciones de Diseño e Implementación:** -La alta,baja y actualización de una cancha solo puede llevarse a cabo si el usuario que intenta realizar la operación tiene permisos de propietario y además ha registrado previamente algun complejo deportivo .

Id	16
Nombre	Como usuario quiero buscar canchas
Descripción	Se permitirá a los buscar diferentes canchas asociadas a complejos deportivos.
Estimación	
Valor para el cliente	
Condiciones de Satisfacción	Al finalizar la operación, los cambios solicitados se ven reflejados en el sistema.
Rendimiento	**Restricciones de Diseño e Implementación:** -La busqueda se realizará a partir de un patrón de busqueda.

Id	17
Nombre	Como usuario quiero poder CRUD reservaciones de turnos a canchas
Descripción	Se permitirá a los usuarios gestionar reservaciones de turnos a canchas
Estimación	
Valor para el cliente	
Condiciones de Satisfacción	Al finalizar la operación, los cambios solicitados se ven reflejados en el sistema.
Rendimiento	**Restricciones de Diseño e Implementación:** -El usuario no puede ser invitado para acceder a la seccion de reservaciones. -El usuario común tiene permisos de crear,listar,actualizar y eliminar sus propias reservaciones de turnos a canchas. -El usuario propietario tiene permisos de listar,actualizar y eliminar cualquier reservación asociada a alguno de sus complejos. Como así también registrar reservaciones y asociar a un usuario común determinado de MG.

Id	18
Nombre	Como usuario quiero CRUD equipos
Descripción	Se permitirá a los usuarios gestionar equipos
Estimación	
Valor para el cliente	
Condiciones de Satisfacción	Al finalizar la operación, los cambios solicitados se ven reflejados en el sistema.
Rendimiento	**Restricciones de Diseño e Implementación:** -El usuario común tiene los permisos necesarios para CRUD equipos -El capitán de un equipo creado es quien lo registra, aunque puede cambiarse el capitán actualizando el equipo. -El único que puede actualizar o eliminar un equipo es el capitán del mismo. -El usuario propietario solo tiene permiso de visualización de equipos, como también lo tiene un usuario invitado.

Id	19
Nombre	Como usuario quiero poder CRUD torneos
Descripción	Se permitirá a los usuarios gestionar torneos
Estimación	
Valor para el cliente	
Condiciones de Satisfacción	Al finalizar la operación, los cambios solicitados se ven reflejados en el sistema.
Rendimiento	**Restricciones de Diseño e Implementación:** -Unicamente podrá crear,actualizar y eliminar torneos los usuarios con permisos de propietario, además debe contar con algún complejo registrado previamente para indicar donde se disputa. -Los usuarios que poseen los permisos de usuarios común o usuarios propietarios tendran acceso al detalle de la visualización de torneo o de la busqueda de torneos.

Id	20
Nombre	Como usuario quiero poder CRUD fixtures
Descripción	Se permitirá a los usuarios gestionar fixtures.
Estimación	
Valor para el cliente	
Condiciones de Satisfacción	Al finalizar la operación, los cambios solicitados se ven reflejados en el sistema.
Rendimiento	**Restricciones de Diseño e Implementación:** -El usuario identificado tiene los permisos necesarios para crear,consultar,actualizar y eliminar fixtures donde tiene permisos de propietario. -Un usuario común solo puede consultar los fixtures registrados. -Un usuario invitado no tiene acceso a los fixtures.

Id	21
Nombre	Como usuario quiero poder CRUD partidos en fixtures
Descripción	Se permitirá a los usuarios gestionar partidos correspondientes a un determinado fixture asociado a un torneo
Estimación	
Valor para el cliente	
Condiciones de Satisfacción	Al finalizar la operación, los cambios solicitados se ven reflejados en el sistema.
Rendimiento	**Restricciones de Diseño e Implementación:** -El usuario que tiene permisos unicamente para CRUD es aquel que tiene los permisos de usuario propietario -Un usuario común tiene permisos unicamente para la consulta de partidos . -Un usuario invitado tiene permiso para realizar la búsqueda de partidos. -Al momento de la creación de un partido debe contarse con un fixture creado,ademas debe asegurarse de que el elequipo visitante sea distinto al local. También debe asegurarse que la fecha del partido sea mayor o igual a fecha de inicio del torneo en cuestión.

Id	22
Nombre	Como usuario quiero poder CRUD de estadisticas de jugadores
Descripción	Se permitirá a los usuarios gestionar estadisticas(goles,tarjetas amarillas y tarjetas rojas) de jugadores a torneos disputados.
Estimación	
Valor para el cliente	
Condiciones de Satisfacción	Al finalizar la operación, los cambios solicitados se ven reflejados en el sistema.
Rendimiento	**Restricciones de Diseño e Implementación:** -El usuario habilitado para realizar las tareas de CRUD de estadisticas es todo usuario con permisos de usuario propietario, en cual debe ser el propietario del complejo en cual se disputa el torneo en el que participa el juuegador. -Todo usuario invitado o usuario comun puede acceder a la visualización de la informacion estadística de un jugador.

Nota: Todo usuario con permisos de administrador tendra los permisos necesarios para realizar CRUD en uno de los modulos que compone el sistema MG, lo cual es coherente pues tiene permiso a sistema en sus totalidad.

3.3 -Requisitos no funcionales

3.3.1- Requisitos de rendimiento

Garantizar que el diseño de las consultas u otro proceso no afecte el desempeño de la base de datos, ni considerablemente el tráfico de la red.

3.3.2-Seguridad

- Se garantiza la confiabilidad, la seguridad y el desempeño del sistema a los diferentes usuarios. En este sentido la información almacenada o registros realizados podrán ser consultados y actualizados permanente.

- Se garantiza la seguridad del sistema con respecto a la información y datos que se manejan tales como archivos y contraseñas. Con respecto respecto a las contraseñas,cada contraseña es sometida a una encriptación para poder brindar mayor seguridad.

3.3.3-Fiabilidad

- El sistema posee una interfaz de uso intuitiva y sencilla.
- La interfaz de usuario se ajusta a las características de la web de los distintos dispositivos de acceso (computadoras de escritorio, computadoras portátiles, tablets, smartphones).

3.3.4-Disponibilidad

- La disponibilidad del sistema es 7 dias a la semana durante las 24.

3.3.5-Mantenibilidad

• El sistema cuenta de una documentación fácilmente actualizable que permita realizar operaciones de mantenimiento con el menor esfuerzo posible.

3.3.6-Portabilidad

• El sistema será implantado bajo la plataforma Web, para la cual será necesario contar con dispositivo (pc de escritorio,notebook,smartphone o tablet) con acceso a internet(debe disponer de un explorador web)

4-Diseño

En este capítulo se describirán cuestiones con respecto al diseño del sistema como la metodología utilizada, diagramas uml, como así también el framework django y se tratarán detalles de como se llevó el diseño de MG a estructura que provee el framework utilizado.

4.1- Metodología de desarrollo

La metodología de desarrollo elegida para MG es SCRUM.

4.1.1-SCRUM

Scrum es el nombre con el que se denomina al framework de desarrollo ágil caracterizado por:

- Adoptar una estrategia de desarrollo incremental, en lugar de la planificación y ejecución completa del producto.
- Basar la calidad del resultado más en el conocimiento tácito de las personas en equipos auto-organizados, que en la calidad de los procesos empleados.
- Solapamiento de las diferentes fases del desarrollo, en lugar de realizar una tras otra en un ciclo secuencial o de cascada.

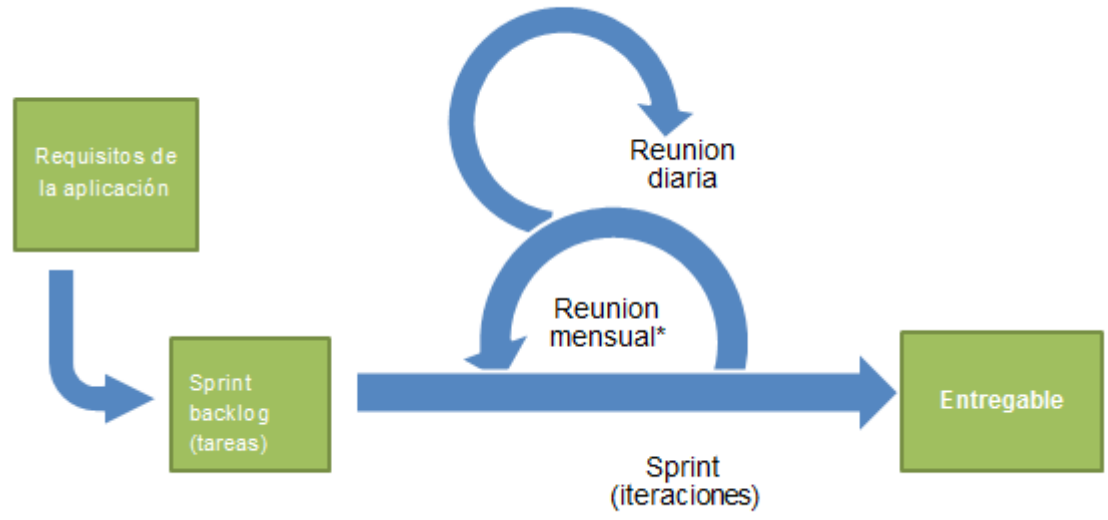


Gráfico del proceso scrum

Características:

SCRUM es un modelo de referencia que define un conjunto de prácticas y roles, y que puede tomarse como punto de partida para definir el proceso de desarrollo que se ejecutará durante un proyecto. Los roles principales en Scrum son el *ScrumMaster*, que procura facilitar la aplicación de scrum y gestionar cambios, el *ProductOwner*, que representa a los *stakeholders* (interesados externos o internos), y el *Team* que ejecuta el desarrollo y demás elementos relacionados con el.

Durante cada *sprint*, un periodo entre una y cuatro semanas (la magnitud es definida por el equipo y debe ser lo mas corta posible), el equipo crea un incremento de software *potencialmente entregable* (utilizable). El conjunto de características que forma parte de cada sprint viene del *Product Backlog*, que es un conjunto de requisitos de alto nivel priorizados que definen el trabajo a realizar. Los elementos del *Product Backlog* que forman parte del sprint se determinan durante la reunión de *Sprint Planning*. Durante esta reunión, el *Product Owner* identifica los elementos del *Product Backlog* que quiere ver completados y los hace del conocimiento del equipo. Entonces, el equipo conversa con el Product Owner buscando claridad y magnitud adecuadas para luego determinar la cantidad de ese trabajo que puede comprometerse a completar durante el siguiente sprint.

Durante el sprint, nadie puede cambiar el Sprint Backlog, lo que significa que los requisitos están congelados durante el sprint.
Scrum permite la creación de equipos auto-organizados impulsando la co-localización de todos los miembros del equipo, y la comunicación verbal entre todos los miembros y disciplinas involucrados en el proyecto.

Un principio clave de Scrum es el reconocimiento de que durante un proyecto los clientes pueden cambiar de idea sobre lo que quieren y necesitan, y que los desafíos impredecibles no pueden ser fácilmente enfrentados de una forma predictiva y planificada. Por lo tanto, Scrum adopta una aproximación pragmática, aceptando que el problema no puede ser completamente entendido o definido, y centrándose en maximizar la capacidad del equipo de entregar rápidamente y responder a requisitos emergentes.

Las características más marcadas que se logran notar en Scrum serían: gestión regular de las expectativas del cliente, resultados anticipados, flexibilidad y adaptación, retorno de inversión, mitigación de riesgos, productividad y calidad, alineamiento entre cliente y equipo, por último equipo motivado. Cada uno de estos puntos mencionados hacen que Scrum sea utilizado de manera regular en un conjunto de buenas prácticas para el trabajo en equipo y de esa manera obtener resultados posibles.

Existen varias implementaciones de sistemas para gestionar el proceso de Scrum, que van desde notas amarillas "post-it" y pizarras hasta paquetes de software. Una de las mayores ventajas de Scrum es que es muy fácil de aprender, y requiere muy poco esfuerzo para comenzarse a utilizar.

4.2-Diagrama UML

El siguiente diagrama es un diagrama UML del sistema en su totalidad.Para ello se desarrollo utilizando el software StartUML (versión 2.0.1).

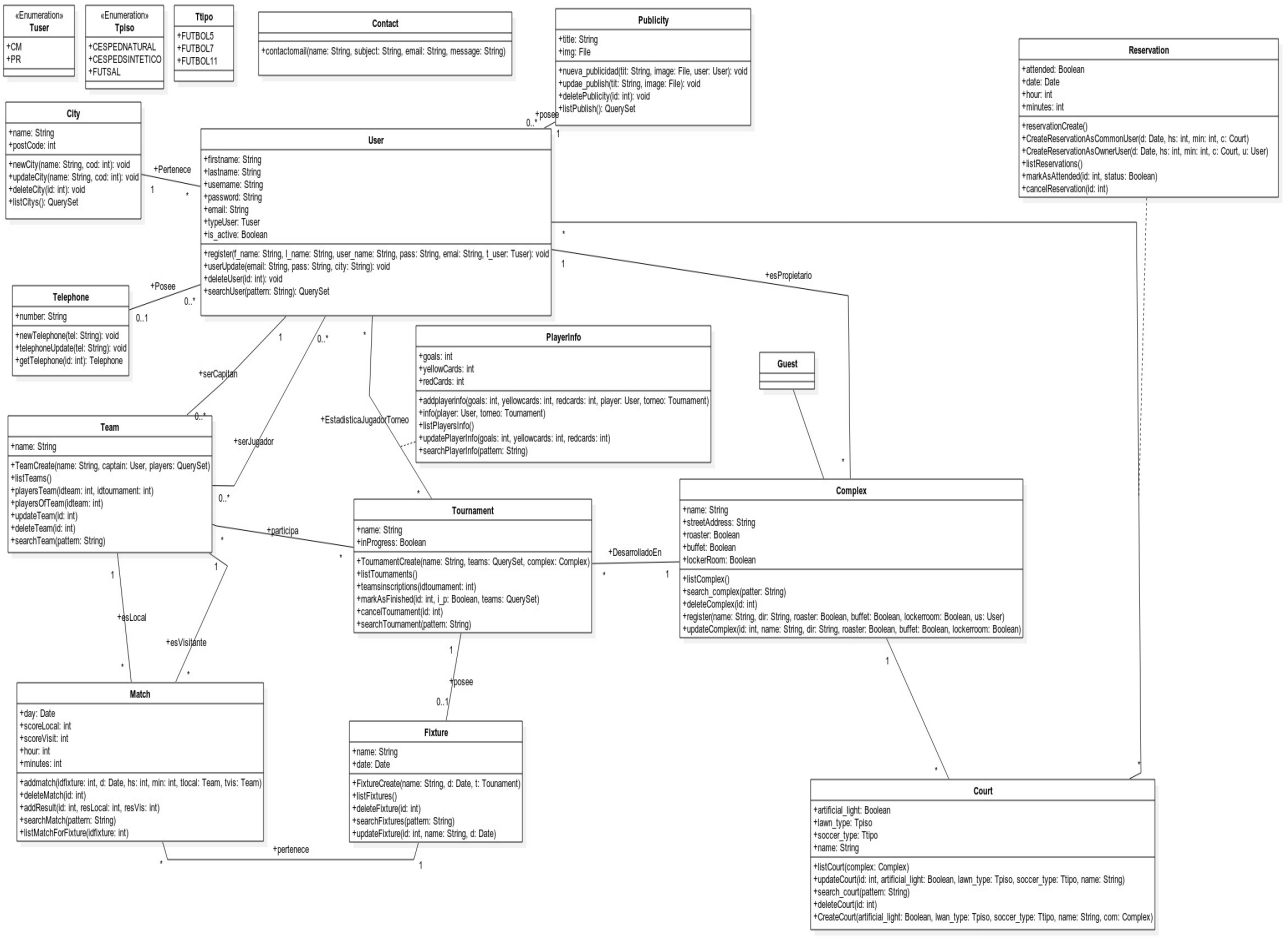


Diagrama de clases

4.3- Diagramas de secuencias

4.3.1-Diagrama de secuencia del proceso para la creación de un equipo

El siguiente diagrama de secuencia muestra el proceso que se lleva a cabo al momento de la creación de un equipo.

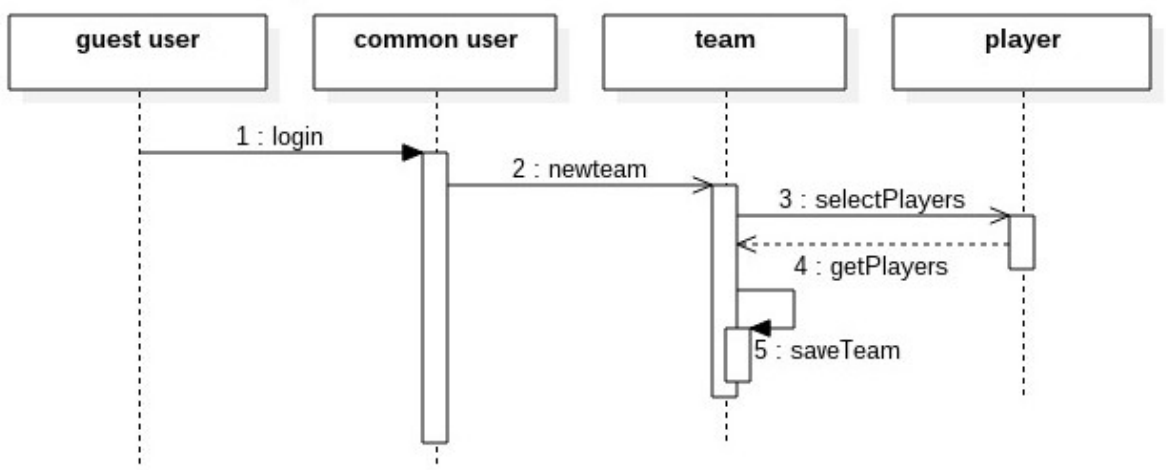


Diagrama de secuencia de la funcionalidad con Id: 18

4.3.2-Diagrama de secuencia del proceso para la creación de un torneo

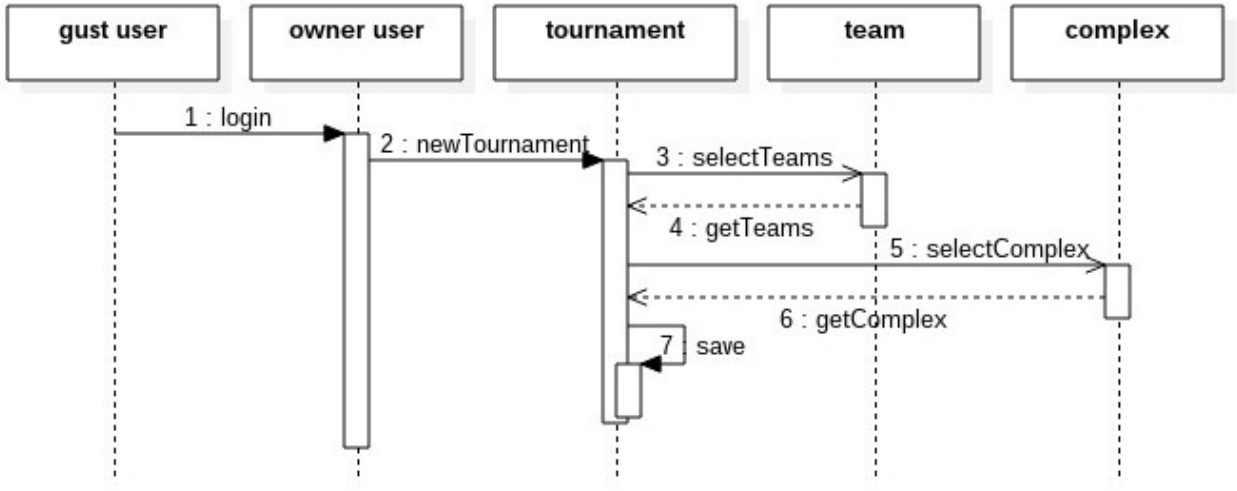


Diagrama de secuencia de la funcionalidad con Id: 19

El siguiente diagrama de secuencia muestra el proceso que se lleva a cabo al momento de la creación de un torneo.

4.3.3-Diagrama de secuencia del proceso para la creación de una reserva

El siguiente diagrama de secuencia muestra el proceso que se lleva a cabo al momento de la creación de un reserva por parte de un usuario UCM.

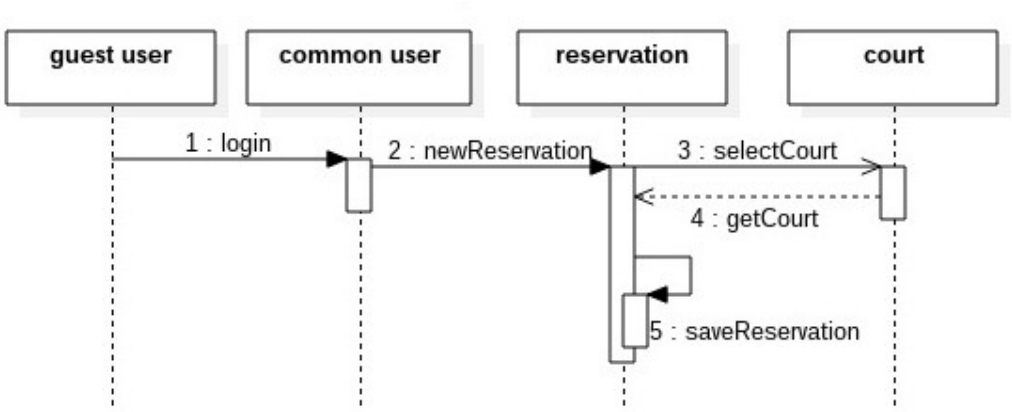


Diagrama de secuencia de la funcionalidad con Id: 17

El siguiente diagrama de secuencia muestra el proceso que se lleva a cabo al momento de la creación de un reserva por parte de un usuario UPR.

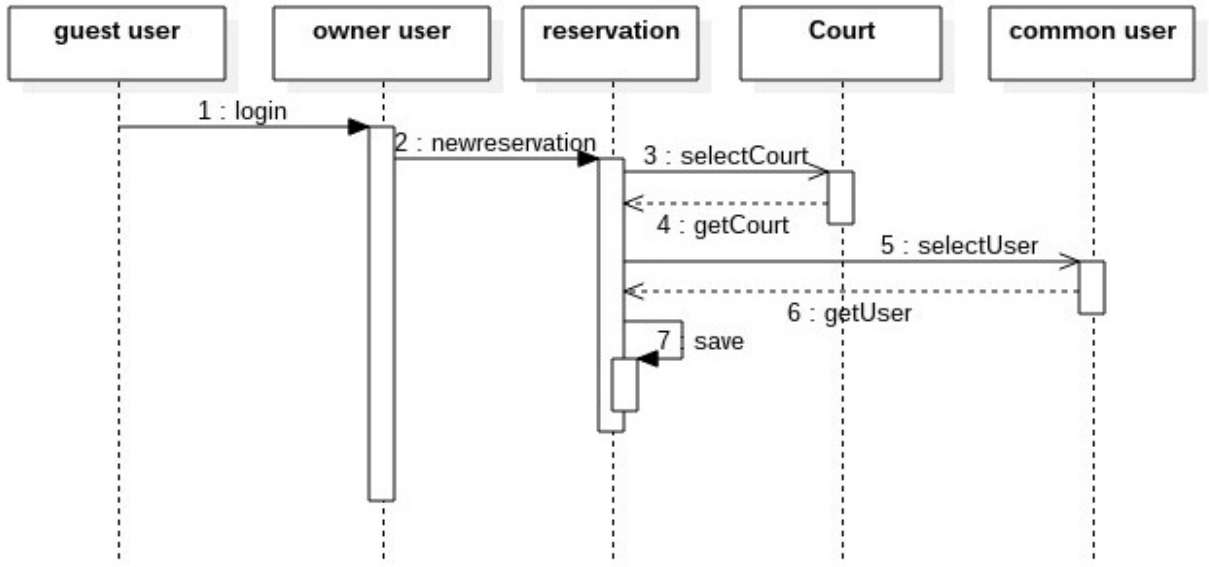


Diagrama de secuencia de la funcionalidad con Id: 17

5-Testing

En este capítulo se dará una introducción del testing en django como así también se describirá el testing realizado en MG.

5.1-Introducción

El testing automatizado es una herramienta extremadamente útil para el desarrollo como también para el desarrollador. Puede utilizarse una colección o conjunto de pruebas para resolver una serie de problemas:

- Cuando se escribe nuevas funcionalidades, puede utilizarse pruebas para validar el código de la funcionalidad para verificar que funciona como se esperaba.

- Cuando se refactoriza o modifica código de funcionalidades existentes, pueden utilizarse las pruebas para asegurarse de que sus cambios no han afectado el comportamiento de su aplicación de forma inesperadas.

La prueba de una aplicación resulta ser una tarea compleja, debido a que una aplicación se compone de varias capas de logica para formar la validación y el procesamiento de información.

La mejor forma de escribir pruebas en Django es usar el modulo **unittest** que viene integrado en la biblioteca estandar de Python.También se puede utilizar cuaquier otro modulo de pruebas Python, Django proporciona su propio modulo de testing para llevar adelante las pruebas automatizadas.

5.1.1-Escribir pruebas

Las pruebas de Django utilizan un módulo de la biblioteca estándar de Python: unittest.Este módulo define pruebas utilizando un enfoque basado en clases. Básicamente Django utiliza una clase **TestCase** (que es subclase de **unittest**) para poder desarrollar pruebas automatizadas. La forma de poder definir pruebas automatizadas es extender la clase TestCase de la siguiente manera :

```
from django.test import TestCase
from myapp.model import animals

class MyAppTestCase(TestCase):
        def  setUp(def):
                animals.objects.create(nombre=”gato”,numPies=4)
        def  test_animals_getnameCorrect(self):
                animal = animals.objects.get(numPies=4)
                self.assertTrue(animal.nombre==”gato”)
```

Este es un claro ejemplo de como definir pruebas automatizadas a la aplicación animals, donde MyAppTestCase extiende las funcionalidades TestCase, el método setUp (la utilización de este metodo es opcional, solo debería utilizarse en el caso de ser necesario) se encarga de realizar las inicializaciones necesarias y la prueba test_animals_getnameCorrect verifica si el nombre del animal es correcto

5.1.2-Ejecución de pruebas automatizadas

Existen varias formas de ejecutar las pruebas automatizadas que provee Django sobre el modulo TestCase:

- Para poder ejecutar todas las pruebas que se definen se deberá ejecutar:
  python manage.py test

- Para poder ejecutar todas las pruebas de un determinado modulo se deberá ejecutar:

  Por ejemplo para poder ejecutar todos las pruebas del módulo animals:

  python manage.py test animals

- Para poder ejecutar una determinada prueba automatizada correspondiente a un determinado módulo se deberá ejecutar :

  Por ejemplo para ejecutar la prueba definida en animals:

  python manage.py test animals.test.MyAppTestCase.test_animals_getnameCorrect

5.1.3-La base de datos de prueba

Las pruebas que requieren una base de datos no utilizan la base de datos utilizada en producción sino que se utiliza una base de datos en blanco e independientemente de si las pruebas pasan o no el test exitosamente, las bases de datos de prueba se destruyen cuando se han ejecutado todas las pruebas.

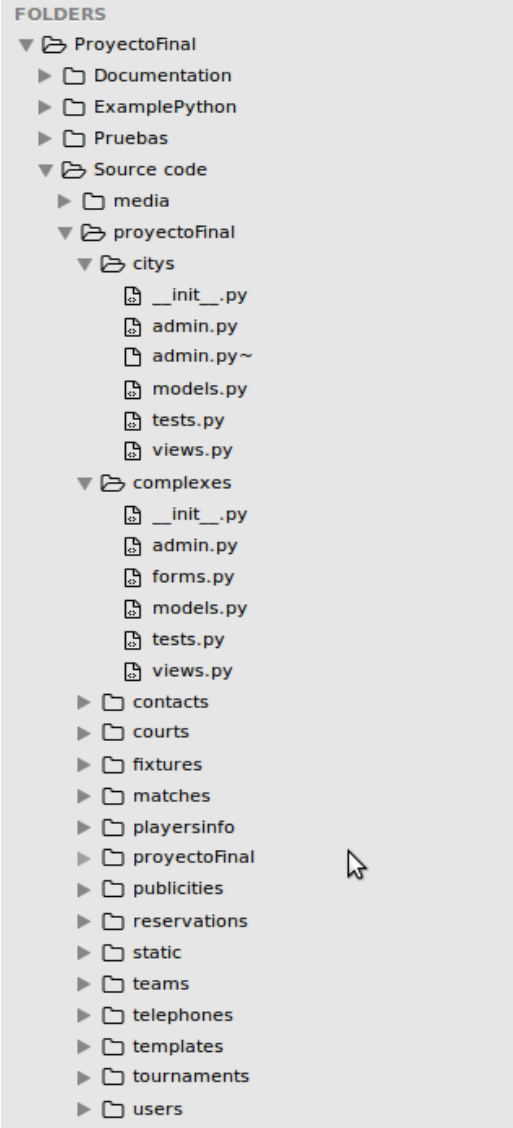
5.1.4-Orden en el que se ejecutan las pruebas

Con el objetivo de garantizar que todo el código de TestCase comienza con una base de datos limpia, el corredor de pruebas Django reordena las pruebas de la siguiente forma:

- Todas las subclases de TestCase se ejecutan primero
- Luego, todos los demas unittests (incluyendo unittest.TestCase) se ejecutan sin ningun orden particular garantizado ni forzada entre ellos.

5.2-Testing en MG

Para realizar el testing de MG durante todo el desarrollo se utilizado el modulo que tiene integrado Django por defecto y para ejecutar las pruebas se ha utilizado el comando *python manage.py test* para correr todos los test. Las pruebas definidas se encuentarn en cada archivo *test.py* de cada aplicación o módulo que compone el sistema como se muestra en la siguiente figura:



Estructura del directorio del proyecto

A continuación se describirán los test o pruebas realizadas:

módulo complexes

Nombre del test	Datos suministrados	Errores	Estado	Solución al error
test_complexes_register	Se genera el test pasando un usuario creado que es un usuario-propietario cuyo username= 'tavo' password='tavo'	No se detectaron errores	Exitoso	Al no detectar errores se debió solucionar ningún error.
test_complexes_update	Se genera el test pasando un usuario creado que es un usuario-propietario cuyo username= 'tavo' password='tavo' y un complejo creado para ser actualizado cuyos datos son name='Oxigeno', streetAddress='Cerrito1159', roaster=True, buffet=True, lockerRoom=True ,user=usuario provisto	No se detectaron errores	Exitoso	Al no detectar errores no se debió solucionar ningún error.
test_complexes_delete	Se genera el test pasando un usuario creado que es un usuario-propietario cuyo username= 'tavo' password='tavo' y un complejo creado para ser	No se detectaron errores	Exitoso	Al no detectar errores se debió solucionar ningún error.

	actualizado cuyos datos son name='Oxigeno', streetAddress='Cerrito1159', roaster=True, buffet=True, lockerRoom=True ,user=usuario provisto			
test_search_complex	Se creo un complejo para poder detectar si realizaba correctamente la busqueda	No se detectaron errores	Exitoso	Al no detectar errores no se debió solucionar ningún error.
test_list_complexes	Se crearon 2 complejos para poder listar complejos y se pasaron 2 usuarios ,un usuario UCM y un usuario UPR	No se detectaron errores	Exitoso	Al no detectar errores no se debió solucionar ningún error.

módulo contacts

Nombre del test	Datos suministrados	Errores	Estado	Solución al error
test_send_email	Se pasaron el asunto del mensaje, el mesaje,quien envía el mensaje y quien es el destinatario,y ademas	No se encontraron errores	Exitoso	Al no detectar errores no se debió solucionar ningún error.

módulo courts

Nombre del test	Datos suministrados	Errores	Estado	Solución al error
test_courts_create	Se suministró un usuario cuyo username='tavo' y password='tavo' y un complejo cuyo name='Oxigeno' y streetAddress='Cerrito 1159'	No se encontraron errores	Exitoso	Al no detectar errores no se debió solucionar ningún error.
test_courts_update	Se suminitaron los siguientes datos: un complejo con name = 'Oxigeno', streetAddress = 'Cerrito 1159',se creo una cancha a actualizar y un usuario cuyo username='tavo' y password='tavo'	No se encontraron errores	Exitoso	Al no detectar errores no se debió solucionar ningún error.
test_courts_delete	Se suminitaron los siguientes datos: un complejo con name = 'Oxigeno', streetAddress = 'Cerrito 1159',se creo una cancha a actualizar y un usuario cuyo username='tavo' y password='tavo'	No se encontraron errores	Exitoso	Al no detectar errores no se debió solucionar ningún error.
test_courts_list	Se crearon 3 canchas para poder realizar el listado de canchas y dos usuarios para verificar acorde al tipo de usuario se debera	No se encontraron errores	Exitoso	Al no detectar errores no se debió solucionar ningún error.

	listar ciertas canchas.			
test_courts_search	Se crearon 3 canchas para poder realizar la busqueda	No se encontraron errores	Exitoso	Al no detectar errores no se debió solucionar ningún error.

módulo fixtures

Nombre del test	Datos suministrados	Errores	Estado	Solución al error
test_fixtures_create	Los datos suministrados son un usuario cuyo username='cris' y password='cris' y un torneo al que se le va asociar un fixture.	No se encontraron errores	Exitoso	Al no detectar errores no se debió solucionar ningún error.
test_fixtures_list	Los datos suministrados son un usuario UCM y un usuario UPR	No se encontraron errores	Exitoso	Al no detectar errores no se debió solucionar ningún error.
test_fixtures_delete	Los datos suministrados son un usuario cuyo username='cris' y password='cris' y un torneo al que se le va asociar un fixture que va a ser creado para luego poder eliminarlo.	Se encontró un error	Falló	Al eliminar un fixture un torneo queda sin fixture y quedaba sin la posibilidad de tener un fixture asociado. El problema se solucionó, al tener todo torneo un fixture asociado en lugar de eliminar un fixture, se elimina el torneo asociado sin que quede el torneo sin referencia a un fixture.

test_fixtures_update	Los datos suministrados son un usuario cuyo username='cris' y password='cris' y un torneo al que se le va asociar un fixture que va a ser creado para luego poder actualizarlo.	No se encontraron errores	Exitoso	Al no detectar errores no se debió solucionar ningún error.
test_fixtures_search	Se le suministra un torneo para crear un fixture para luego ser buscado	No se encontraron errores	Exitoso	Al no detectar errores no se debió solucionar ningún error.

módulo matches

Nombre del test	Datos suministrados	Errores	Estado	Solución al error
test_matches_create	Los datos que se suministran es un usuario cuyo username='cris' y password='cris' y un fixture al que va a pertenecer el partido	No se encontraron errores	Exitoso	Al no detectar errores no se debió solucionar ningún error.
test_matches_delete	Los datos que se suministran es un usuario cuyo username='cris' y password='cris' y un fixture al que va a pertenecer el partido y se crea un partido temporalmente para comprobar que elimina correctamente	No se encontraron errores	Exitoso	Al no detectar errores no se debió solucionar ningún error.

test_matches_update	Los datos que se suministran es un usuario cuyo username='cris' y password='cris' y un fixture al que va a pertenecer el partido y se crea un partido temporalmente para comprobar que elimina correctamente	No se encontraron errores	Exitoso	Al no detectar errores no se debió solucionar ningún error.
test_matches_list	Se le suministra un usuario	No se encontraron errores	Exitoso	Al no detectar errores no se debió solucionar ningún error.
test_matches_search	Se suministran algunos partidos para realizar una busqueda	No se encontraron errores	Exitoso	Al no detectar errores no se debió solucionar ningún error.

módulo playersinfo

Nombre del test	Datos suministrados	Errores	Estado	Solución al error
test_playersinfo_create	Un usuario que representa al usuario logueado, un torneo y el usuario que representa al jugador	Se encontró un error	Falló	Se solucionó verificando si el torneo es de propiedad del usuario logueado
test_playersinfo_list	Un usuario UCM y un usuario UPR	No se encontraron errores	Exitoso	Al no detectar errores no se debió solucionar ningún error.
test_playersinfo_update	Se crea una estadistica para ser actualizada(esto implica suministrar un	No se encontraron errores	Exitoso	Al no detectar errores no se debió solucionar ningún error.

	usuario logueado, un juegador y un torneo)			
test_playersinfo_search	Un usuario UCM y un usuario UPR	No se encontraron errores	Exitoso	Al no detectar errores no se debió solucionar ningún error.

módulo publicities

Nombre del test	Datos suministrados	Errores	Estado	Solución al error
test_publicities_create	Los datos suministrados son el usuario logueado,el nombre de la publicidad y la imagen	No se encontraron errores	Exitoso	Al no detectar errores no se debió solucionar ningún error.
test_publicities_update	Los datos suministrados son el usuario logueado y la publicidad a actualizar	No se encontraron errores	Exitoso	Al no detectar errores no se debió solucionar ningún error.
test_publicities_delete	Los datos suministrados son el usuario logueado y la publicidad a actualizar	No se encontraron errores	Exitoso	Al no detectar errores no se debió solucionar ningún error.
test_publicities_list	Los datos suministrados son el usuario logueado	No se encontraron errores	Exitoso	Al no detectar errores no se debió solucionar ningún error.

módulo reservations

Nombre del test	Datos suministrados	Errores	Estado	Solución al error
test_reservations_create	Los datos suministrados son el usuario logueado ,la fecha,hora, minutos,cancha y en el caso de que el usuario logueado es UPR se suministra un usuario a quien va dirigida la reserva	No se encontraron errores	Exitoso	Al no detectar errores no se debió solucionar ningún error.
test_reservations_list	Los datos suministrados son el usuario logueado	No se encontraron errores	Exitoso	Al no detectar errores no se debió solucionar ningún error.
test_reservations_update	Los datos suministrados son el usuario logueado y un reservación	No se encontraron errores	Exitoso	Al no detectar errores no se debió solucionar ningún error.
test_reservations_cancel	Los datos suministrados son el usuario logueado y un reservación	No se encontraron errores	Exitoso	Al no detectar errores no se debió solucionar ningún error.

módulo teams

Nombre del test	Datos suministrados	Errores	Estado	Solución al error
test_teams_create	Los datos suministrados los jugadores que conforman parte del equipo y el capitán del mismo que viene sería el usuario logueado	No se encontraron errores	Exitoso	Al no detectar errores no se debió solucionar ningún error.
test_teams_list	Ninguno,pues se hace una inicialización pero solo es necesaria para mostrar los equipos en caso de que existieran.	No se encontraron errores	Exitoso	Al no detectar errores no se debió solucionar ningún error.
test_teams_update	Los datos suministrados son el usuario logueado, el equipo a actualizar y la información que se desea actualizar del equipo	No se encontraron errores	Exitoso	Al no detectar errores no se debió solucionar ningún error.
test_teams_delete	Los datos suministrados son el usuario logueado, el equipo a eliminar	No se encontraron errores	Exitoso	Al no detectar errores no se debió solucionar ningún error.
test_teams_search	Los datos suministrados son el nombre del equipo a buscar	No se encontraron errores	Exitoso	Al no detectar errores no se debió solucionar ningún error.

módulo tournaments

Nombre del test	Datos suministrados	Errores	Estado	Solución al error
test_tournaments_create	Los datos suministrados son el usuario logueado, los equipos participantes, y la información pertinente del torneo en cuestión	No se encontraron errores	Exitoso	Al no detectar errores no se debió solucionar ningún error.
test_tournaments_list	Los datos suministrados son el usuario logueado	No se encontraron errores	Exitoso	Al no detectar errores no se debió solucionar ningún error.
test_tournaments_markAsFinished	Los datos suministrados son el usuario logueado, el torneo a actualizar y la información a actualizar(puede ser equipos a agregar en el torneo o indicar la finalizacion o retomar torneo)	No se encontraron errores	Exitoso	Al no detectar errores no se debió solucionar ningún error.
test_tournaments_cancel	Los datos suministrados son el usuario logueado, el torneo a eliminar	No se encontraron errores	Exitoso	Al no detectar errores no se debió solucionar ningún error.
test_tournaments_search	Los datos suministrados son el nombre del torneo	No se encontraron errores	Exitoso	Al no detectar errores no se debió solucionar ningún error.

módulo users

Nombre del test	Datos suministrados	Errores	Estado	Solución al error
test_users_register	Los datos suministrados son ciudad, teléfono, nombre, apellido, nombre de usuario, contraseña, email, tipo de usuario	No se encontraron errores	Exitoso	Al no detectar errores no se debió solucionar ningún error.
test_users_delete	Los datos suministrados son el usuario logueado y el usuario a eliminar	No se encontraron errores	Exitoso	Al no detectar errores no se debió solucionar ningún error.
test_users_userUpdate	Los datos suministrados son el usuario logueado y el usuario a actualizar (perfil).	No se encontraron errores	Exitoso	Al no detectar errores no se debió solucionar ningún error.
test_users_telephoneUpdate	Los datos suministrados son el usuario logueado y el telefono del usuario a actualizar.	No se encontraron errores	Exitoso	Al no detectar errores no se debió solucionar ningún error.

Los errores que se han encontrados en algunos de los módulos es el siguiente :

En aquellos modulos que se definieron algunas de las vistas, utilizando vistas basadas en clases, en dichos casos se utilizaba el método *get_objetc* y ante alguna condición invalida se redirigia utilizando *HttpResponseRedirect.*Lo cual es un fallo, pues en este método se debe tratar únicamente el objeto en si.

Solución: El tratamiento de condiciones inválidas por cuestiones de permisos se trataron en el método *dispatch* en lugar de tratarlas en *get_object*. En cuanto de situaciones especiales particulares del objeto si no se cumple las condiciones necesarias se lanza un error 404.

6- Conclusión y dificultades encontradas

En este capitulo se concluye el proyecto y se detallan las dificultades encontradas a lo largo del desarrollo de **minutogol**.

6.1-Conclusión

Como se puede apreciar a lo largo de este documento, se ha desarrollado un producto en cual se ha podido aplicar diferentes tecnologias aprendidas durante el transcurso de la carrera de analista en computación aplicando tecnologias, buenas prácticas de desarrollo y mantenimiento.

Por otro lado se ha podido desarrollar un producto acorde a los estandares de cálidad caracterizado por la eficiencia y la simplicidad de su uso, facilitando los procesos de reservaciones de turnos a canchas correspondientes a diferentes complejos cómo así también facilitando la administración de complejos deportivos permitiendo gestionar publicidades, torneos, fixtures, partidos y estadisticas.

6.2-Dificultades encontradas

Como dificultad se podría mencionar el tiempo invertido en la instalación de las herramientas y configuración del proyecto, aunque la dificultad más grande fue la de aprender una tecnología nueva como es el framework **Django** ya que implicó aprender primero las bases de python y luego aprender desde lo básico a utilizar el framework.

Si bien existe mucha bibliografía en internet desde manuales, tutoriales, foros y una comunidad muy grande a la hora de llevar a cabo un proyecto del tamaño de **MG**, se debe invertirle el tiempo necesario para aprender en profundidad cada uno de los aspecto que componen al desarrollo web utilizando ésta tecnología.

7-Futuras extensiones

En este capítulo se mencionarán las futuras extensiones que no han sido implementadas debido a las limitaciones de tiempo existentes.
Sería muy interesante seguir con la implementación de nuevas caracteristicas que mejoren la experiencia a los usuarios.

7.1-Lista de futuras extensiones

- Se pretende extender **minutogol** a otras ciudades.

- Permitir a los propietarios de complejos incorporar otras disciplinas deportivas (handball, tenis, basketball, etc...).

- Permitir a usuarios que no son propietarios postularse para equipos cuando hay cupos libres.

- Permitir asignar en que cancha se juega un partido correspondiente al fixture de un determinado torneo.

- Integrar **minutogol** con un determinado sistema de meteorología para que al momento de reservar un turno en caso de que el clima sea inestable me liste las canchas con techo cubierto.

Apendice: Django

Introduccion a Django

Django, es un framework de desarrollo Web que ahorra tiempo y hace que el desarrollo Web sea ágil. Utilizando Django se puede crear y mantener aplicaciones Web de alta calidad con un mínimo esfuerzo.

En el mejor de los casos, el desarrollo web es un acto creativo; en el peor, puede ser una molestia repetitiva y frustrante. Django te permite enfocarte en la parte creativa al mismo tiempo que mitiga el esfuerzo de las partes repetitivas. De esta forma, provee un alto nivel de abstracción de patrones comunes en el desarrollo Web, atajos para tareas frecuentes de programación y convenciones claras sobre cómo solucionar problemas.

Al mismo tiempo, Django intenta no entrometerse, dejándo trabajar fuera del ámbito del framework según sea necesario. El objetivo es incursionar en Django. El enfoque es doble. Primero, explicando en profundidad lo que hace Django, y cómo crear aplicaciones Web con él. Segundo, discutiendo conceptos de alto nivel cuando se considere apropiado.

Al conocer Django, se aprenderá las habilidades necesarias para desarrollar sitios Web poderosos de forma rápida, con código limpio y de fácil mantenimiento.

¿Que es un framework?

Django es un miembro importante de una nueva generación de frameworks Web.

¿Y qué significa ese término exactamente?

Para contestar esa pregunta, consideremos el diseño de una aplicación Web escrita usando el estándar Common Gateway Interface (CGI), una forma popular de desarrollar aplicaciones Web alrededor del año 1998. En esa época, cuando se desarrollaba una aplicación CGI, se hacia todo desde cero.

Por ejemplo,aquí hay un script CGI sencillo, escrito en Python, que muestra los diez libros más recientemente publicados de una base de datos:

```
#!/usr/bin/python

import MySQLdb

print "Content-Type: text/html"
print
print "<html><head><title>Libros</title></head>"
print
print "<body>"
print "<h1>Los ultimos 10 libros</h1>"
print "<ul>"
conexion = MySQLdb.connect(user='yo', passwd='dejame_entrar', db='mi_base')
cursor = conexion.cursor()
cursor.execute("SELECT nombre FROM libros ORDER BY fecha_pub DESC LIMIT 10")
for fila in cursor.fetchall():
print "<li> %s</li>" % fila[0]
print "</ul>"
print "</body></html>"
conexion.close()
```

Este código es fácil de entender. Primero imprime una línea de #Content-Type#, seguido de una línea en blanco, tal como requiere CGI. Imprime HTML introductorio, se conecta a la base de datos y ejecuta una consulta que obtiene los diez libros más recientes. Hace un bucle sobre esos libros y genera una lista HTML desordenada. Finalmente imprime el código para cerrar el HTML y cierra la conexión con la base de datos.

Con una única página dinámica como esta, el enfoque desde cero no es necesariamente malo. Por un lado, este código es sencillo de comprender -- incluso un desarrollador novato puede leer estas 16 líneas de Python y entender todo lo que hace--. No hay más nada que aprender; no hay más código para leer. También es sencillo de utilizar: tan sólo guarda este código en un archivo llamado ultimoslibros.cgi, sube ese archivo a un servidor Web y visita esa página con un navegador.

Pero a medida que una aplicación Web crece más allá de lo trivial, este enfoque se desmorona y te enfrentas a una serie de problemas:

¿Qué sucede cuando múltiples páginas necesitan conectarse a la base de datos?

Seguro que ese código de conexión a la base de datos no debería estar duplicado en cada uno de los scripts CGI, así que la forma pragmática de hacerlo sería refactorizarlo en una función compartida.

¿Debería un desarrollador realmente tener que preocuparse por imprimir la línea de #Content-Type# y acordarse de cerrar la conexión con la base de datos?

Este tipo de código repetitivo reduce la productividad del programador e introduce la oportunidad para que se cometan errores. Estas tareas de configuración y cierre estarían mejor manejadas por una infraestructura común.

¿Qué sucede cuando este código es reutilizado en múltiples entornos, cada uno con una base de datos y contraseñas diferentes?

En ese punto, se vuelve esencial alguna configuración específica del entorno.

¿Qué sucede cuando un diseñador Web que no tiene experiencia programando en Python desea rediseñar la página?

Lo ideal sería que la lógica de la página -- la búsqueda de libros en la base de datos -- esté separada del código HTML de la página, de modo que el diseñador pueda hacer modificaciones sin afectar la búsqueda.

En si, un framework web es un software que alivia el sufrimiento derivado de construir páginas web dinámicas. Abstrae problemas comunes al desarrollo web y proporciona atajos para tareas de programación frecuentes.

Para los lectores que desconocen el desarrollo de sitios web dinámicos: un sitio web dinámico es uno en el que las páginas no son simplemente documentos HTML colocados en algún lugar del sistema de ficheros de un servidor.

En cambio, en un sitio web dinámico, cada página la genera un programa de computador —una famosa "aplicación web"— que usted, el desarrollador web, crea. Por ejemplo, una aplicación web podría obtener registros de una base de datos o realizar alguna acción basándose en la entrada del usuario.

Un buen framework web resuelve los siguientes problemas:

- Proporciona un método para hacer corresponder peticiones URL con el código que maneja las peticiones. En otras palabras, le otorga una manera de designar qué código se ejecutará para cada URL. Por ejemplo, se le podría decir al framework: "Para las URLs que se parezcan a /users/joe/, ejecuta el código que muestre el perfil del usuario con ese nombre de usuario".

- Facilitar mostrar, validar y volver a mostrar formularios HTML. Los formularios HTML son la principal manera de obtener datos de entrada de los usuarios web, así que es mejor que un framework web facilite la representación de formularios y el manejo del código tedioso para mostrar y volver a mostrar formularios (resaltando los errores).

•Convierte la entrada que envía el usuario en estructuras de datos que se pueden manipular cómodamente. Por ejemplo, el framework podría convertir los datos de un formulario HTML en tipos de datos nativos al lenguaje de programación que se esté utilizando.

•Ayuda a separar el contenido de la presentación mediante un sistema de plantillas, de manera que se pueda cambiar el aspecto de un sitio web sin afectar al contenido, y viceversa.

•Se integra cómodamente con las capas de almacenamiento —como las bases de datos— pero no exije estrictamente el uso de una base de datos.

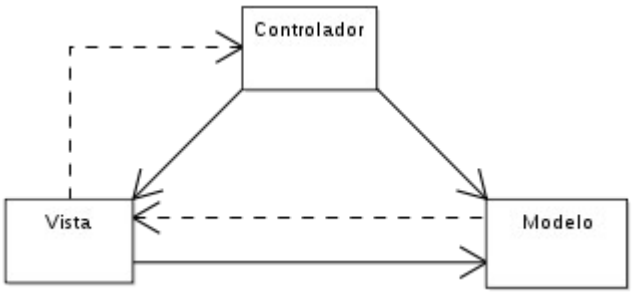
•Le permite trabajar más productivamente, a un nivel de abstracción mayor que si estuviera programando usando, digamos, HTTP. Pero no le prohíbe ir un nivel de astracción "hacia abajo" cuando sea necesario.

•Se aparta de su camino, evitando llenarle la aplicación de manchas sucias, como URLs que contengan ".aspx" o ".php".

Django hace todas estas cosas correctamente; y presenta una serie de características que elevan el listón de lo que debería ser un framework web.
El framework está escrito en Python, un lenguaje de programación bonito, conciso, potente y de alto nivel. Para desarrollar un sitio utilizando Django hay que escribir código Python que utiliza las bibliotecas de Django.

El modelo mvc

El modelo–vista–controlador (MVC) es un patrón de arquitectura de software que separa los datos y la lógica de negocio de una aplicación de la interfaz de usuario y el módulo encargado de gestionar los eventos y las comunicaciones. Para ello MVC propone la construcción de tres componentes distintos que son el modelo, la vista y el controlador, es decir, por un lado define componentes para la representación de la información, y por otro lado para la interacción del usuario.Este patrón de arquitectura de software se basa en las ideas de reutilización de código y la separación de conceptos, características que buscan facilitar la tarea de desarrollo de aplicaciones y su posterior mantenimiento.



Esquema del patrón MVC

El patrón MVC fue una de las primeras ideas en el campo de las interfaces gráficas de usuario y uno de los primeros trabajos en describir e implementar aplicaciones software en términos de sus diferentes funciones.
El MVC fue introducido por Trygve Reenskaug (web personal) en Smalltalk-76 en los años 70 y, seguidamente, en los años 80, Jim Althoff y otros implementaron una versión de MVC para la biblioteca de clases de Smalltalk-80.Sólo más tarde, en 1988, MVC se expresó como un concepto general en un artículo sobre Smalltalk-80.
En esta primera definición de MVC el **controlador** se definía como "*el módulo que se ocupa de la entrada*" (de forma similar a como la **vista** "*se ocupa de la salida*"). Esta definición no tiene cabida en las aplicaciones modernas en las que esta funcionalidad es asumida por una combinación de la 'vista' y algún framework moderno para desarrollo. El 'controlador', en las aplicaciones modernas de la década de 2000, es un módulo o una sección intermedia de código, que hace de intermediario de la comunicación entre el 'modelo' y la 'vista', y unifica la validación (utilizando llamadas directas o el "*observer*" para desacoplar el 'modelo' de la 'vista' en el 'modelo').

De manera genérica, los componentes de MVC se podrían definir como sigue:

•El **Modelo**: Es la representación de la información con la cual el sistema opera, por lo tanto gestiona todos los accesos a dicha información, tanto consultas como actualizaciones, implementando también los privilegios de acceso que se hayan descrito en las especificaciones de la aplicación (lógica de negocio). Envía a la 'vista' aquella parte de la información que en cada momento se le solicita para que sea mostrada (típicamente a un usuario). Las peticiones de acceso o manipulación de información llegan al 'modelo' a través del 'controlador'.
•El **Controlador**: Responde a eventos (usualmente acciones del usuario) e invoca peticiones al 'modelo' cuando se hace alguna solicitud sobre la información (por ejemplo, editar un documento o un registro en una base de datos). También puede enviar comandos a su 'vista' asociada si se solicita un cambio en la forma en que se presenta de 'modelo' (por ejemplo, desplazamiento o scroll por un documento o por los diferentes registros de una base de datos), por tanto se podría decir que el 'controlador' hace de intermediario entre la 'vista' y el 'modelo'.
•La **Vista**: Presenta el 'modelo' (información y *lógica de negocio*) en un formato adecuado para interactuar (usualmente lainterfaz de usuario) por tanto requiere de dicho 'modelo' la información que debe representar como salida.

MVC y bases de datos

Muchos sistemas informáticos utilizan un Sistema de Gestión de Base de Datos para gestionar los datos que debe utilizar la aplicación; en líneas generales del **MVC** dicha gestión corresponde al modelo. La unión entre *capa de presentación* y *capa de negocio* conocido en el paradigma de la Programación por capas representaría la integración entre la **Vista** y su correspondiente **Controlador** de eventos y acceso a datos, MVC no pretende discriminar entre capa de negocio y capa de presentación pero si pretende separar la capa *visual gráfica* de su correspondiente *programación y acceso a datos*, algo que mejora el desarrollo y mantenimiento de la *Vista* y el*Controlador* en paralelo, ya que ambos cumplen ciclos de vida muy distintos entre sí.

Django y el "Patrón MTV"

Cabe destacar que Django fue diseñado para promover el acoplamiento débil y la estricta separación entre las piezas de una aplicación. Siguiendo esta filosofía es fácil hacer cambios en un lugar particular de la aplicación sin afectar otras piezas.

Existen tres piezas (la lógica de acceso a la base de datos, la lógica de negocios, y la lógica de presentación) las cuales, unidas, comprenden un concepto que a veces es llamado el patrón de arquitectura de software Modelo-Vista-Controlador (MVC). En este patrón, el "Modelo" hace referencia al acceso a la capa de datos, la "Vista" se refiere a la parte del sistema que selecciona qué mostrar y cómo mostrarlo, y el "Controlador" implica la parte del sistema que decide qué vista usar, dependiendo de la entrada del usuario, accediendo al modelo si es necesario.

Django sigue el patrón MVC tan al pie de la letra que puede ser llamado un framework MVC. La M, V y C se separan en Django de la siguiente manera:

• M, la porción de acceso a la base de datos, es manejada por la capa de la base de datos de Django.

• V, la porción que selecciona qué datos mostrar y cómo mostrarlos, es manejada por la vista y las plantillas.

• C, la porción que delega a la vista dependiendo de la entrada del usuario, es manejada por el framework mismo siguiendo tu URLconf y llamando a la función apropiada de Python para la URL obtenida. Debido a que la "C" es manejada por el mismo framework y la parte más emocionante se produce en los modelos, las plantillas y las vistas, Django es conocido como un framework MTV.

En el patrón de diseño MTV:

• M significa "Model" (Modelo): El modelo define los datos almacenados, se encuentra en forma de clases de Python, cada tipo de dato que debe ser almacenado se encuentra en una variable con ciertos parámetros, posee métodos también. Todo esto permite indicar y controlar el comportamiento de los datos.

• T significa "Template" (Plantilla), la capa de presentación. Esta capa contiene las decisiones relacionadas a la presentación: es básicamente una página HTML con algunas etiquetas extras propias de Django, eso si no solamente crea contenido en HTML (también XML, CSS, Javascript, CSV, etc).

• V significa "View" (Vista): La vista se presenta en forma de funciones en Python, su propósito es determinar que datos serán visualizados, entre otras cosas más que iremos viendo conforme avanzamos con el curso. El ORM de Django permite escribir código Python en lugar de SQL para hacer las consultas que necesita la vista. La vista también se encarga de tareas conocidas como el envío de correo electrónico, la autenticación con servicios externos y la validación de datos a través de formularios. Lo mas importante a entender con respecto a la vista es que no tiene nada que ver

con el estilo de presentación de los datos, sólo se encarga de los datos, la presentación es tarea de la plantilla.

Además, Django posee un mapeo de URLs que permite controlar el despliegue de las vistas, esta configuración es conocida como URLConf. El trabajo del URLConf es leer la URL que el usuario solicitó, encontrar la vista apropiada para la solicitud y pasar cualquier variable que la vista necesite para completar su trabajo. El URLConf esta construido con expresiones regulares en Python y sigue la filosofia de Python: “Explicito es mejor que implícito”. Este URLConf permite que las rutas que maneje Django sean agradables y entendibles para el usuario.