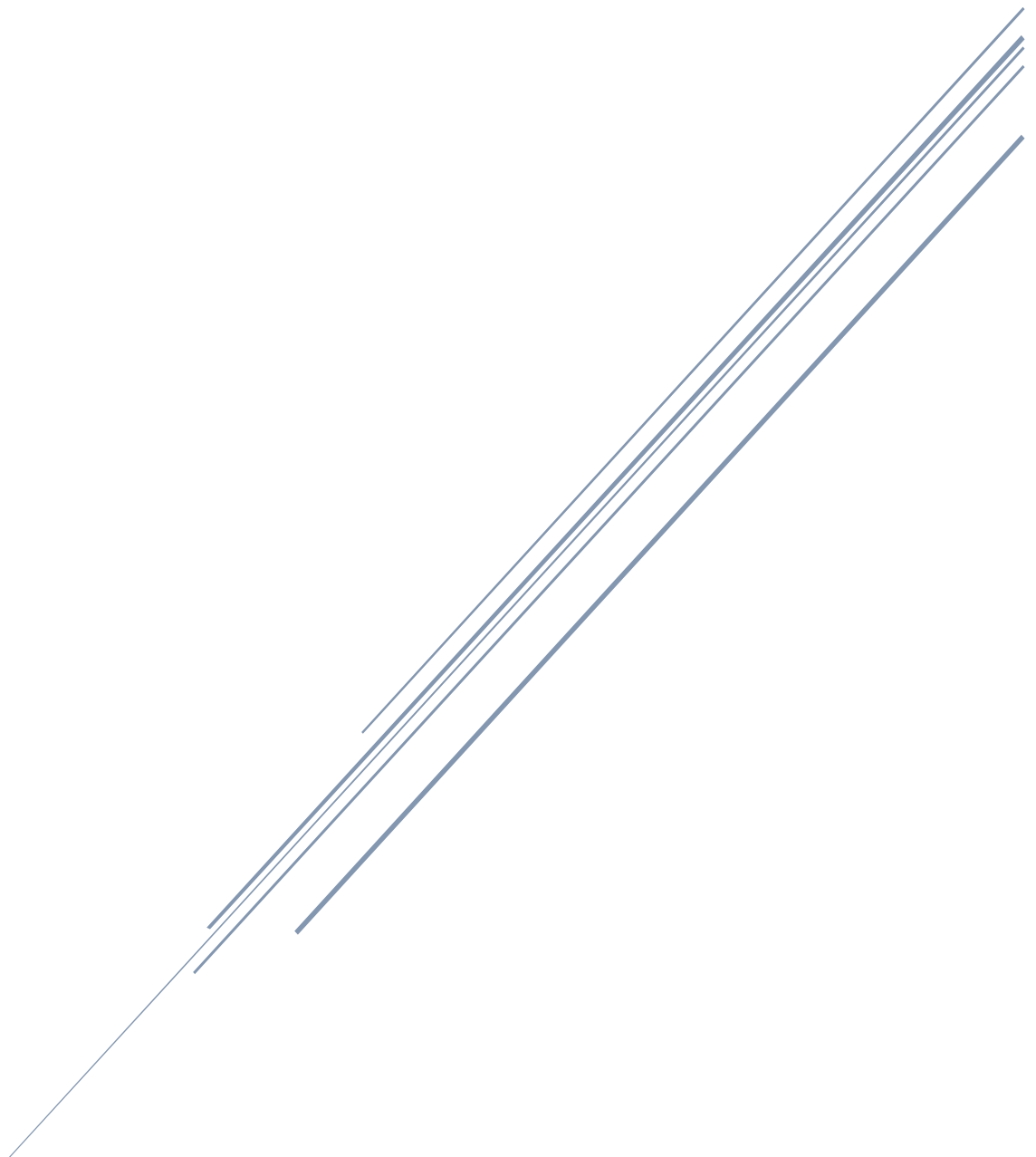


CBGRANDSLAMDJANGO

Carlos Alfredo Cervantes Bedoy



UNIVERSIDAD AUTONOMA DE AGUASCALIENTES
TALLER 2



Dicho proyecto se trata de hacer una simulación del ecosistema de Grand Slam, dicho en otras palabras simularemos la administración de datos que pueden ser manejados dentro de un torneo de Grand Slam, el sistema conceptualiza el manejo de entidades mas representativas del torneo es decir, jugadores, torneros, entrenadores, países, localidades, juegos, modalidades etc.

Dicho sistema facilita la comprensión de como están comunicadas dichas entidades u objetos entre si, para que con un buen diseño de arquitectura y análisis de las entidades se logren obtener resultados esperados y facilitar a los usuarios el hacer cálculos, investigaciones etc en papel.

La descripción del problema es la siguiente:

El sistema debe memorizar todos los encuentros que se han desarrollado desde que existe el torneo, así como las siguientes características de estos.

Descripción:

El Grand Slam se compone de cuatro torneos anuales que se celebran en Gran Bretaña, Estados Unidos, Francia y Australia. En cada país se pueden desarrollar en distintos lugares (p. ej., en EE. UU. Puede desarrollarse en Forest Hill o en Flushing Meadows).

Cada partido tiene asociado un premio de consolación para el perdedor que dependerá de la fase en que se encuentre el torneo (p. ej., el perdedor de octavos de final puede ganar 5.000 dólares). El ganador de al final recibirá el premio correspondiente al torneo.

Cada torneo tiene cinco modalidades: Individual masculino, individual femenino, dobles masculino, dobles femenino y dobles mixtos.

También hay que tener en cuenta la nacionalidad de un jugador, de forma que éste puede ser apátrida o tener varias nacionalidades. Resultados a considerar:

El sistema debe dar respuesta a las siguientes preguntas:

1. Dado un año y un torneo, composición y resultado de los partidos.
2. Lista de árbitros que participaron en el torneo.
3. Ganancias percibidas en premios por un jugador a lo largo del torneo.
4. Lista de entrenadores que han entrenado a un jugador a lo largo del torneo y fechas en las que lo hizo.

Ejemplos de acceso a la base de datos.

1. Connors gana Gerulaitis en Roland Garros en 1979 en cuartos de final en individuales masculinos por 6-3 4-6/7-5 6-0.
2. El señor Wilkinson arbitro ese partido.
3. Alemania ha ganado dos veces las individuales masculinas de Wimbledon.
Borg ha ganado 2.000.000 de dólares a lo largo de su participación en el Grand Slam.
4. El ganador de Roland Garros de 1987 ganó 20.000 dólares.

Posible solución:

La solución parte de la hora en que nos sentamos a analizar la situación del ecosistema ya que este no tiene la parte tecnológica implementada y se deberá partir de implementar, la principal necesidad del cliente o los clientes es que necesitan un sistema agil y escalable en módulos, nosotros como desarrolladores debemos tener una buena arquitectura que pueda solucionar el sistema y a la vez hacerlo escalable en cualquier momento del espacio, es por eso que es necesario implementar una solución que facilite a los usuarios ver y administrar el contenido desde su computadora, Tablet o bien celular.

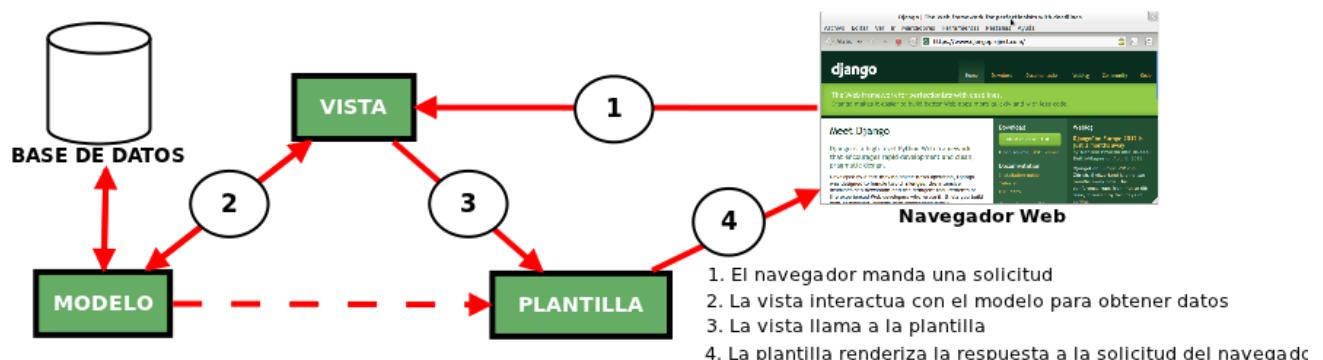
Para ello se desarrollara una aplicación web la cual facilite al usuario la alta, baja, consulta y reportes de la información que se requiera; dicho así el lenguaje de programación que se utilizara será **Python**, un lenguaje de programación dinámico y de fácil comprensión e implementación, para esto se optara por el desarrollo a través el **framework** Django, un **framework** open source enfocado puramente al **modelo controlador template**.

Diseño:

Una breve descripción de esta arquitectura seria:

MODELO	CONTROLADOR	TEMPLATE
En este se definen las entidades de la base de datos, que podría partir de un poco de la lógica del sistema, además de que provee la escalabilidad en cualquier momento del sistema Python – SQL	Este se encarga de la lógica del negocio, si se requiere mostrar una función de alta, baja, consulta, cambio, reporte etc (Backend). Además que se encarga de las validaciones correspondientes Python – Javascript	Este se encarga básicamente del front-end el cual representara la información generada por el backend, así como dar la entrada de datos hacia el backend Python – Javascript - HTML

Workflow



CBGRANDSLAMDJANGO

Ejemplo de modelos (Python ♥)

```
class Player(models.Model):
    sex_l = (
        ('F', 'Female'),
        ('M', 'Male'),
    )
    age_l = (
        (20, '20'),
        (21, '21'),
        (22, '22'),
        (23, '23'),
        (24, '24'),
        (25, '25'),
        (26, '26'),
        (27, '27'),
        (28, '28'),
        (29, '29'),
        (30, '30'),
    )
    firstName = models.CharField('First name', max_length=45)
    lastName = models.CharField('Last name', max_length=45)
    age = models.IntegerField(choices=age_l)
    sex = models.CharField(max_length=10, choices=sex_l)
    height = models.IntegerField(null=True)
    weight = models.IntegerField(null=True)
    web = models.CharField(max_length=45, null=True)
    facebook = models.CharField(max_length=45, null=True)
    email = models.CharField(max_length=45, null=True)
    nationality = models.ForeignKey(Nationality)
    trainer = models.ForeignKey(Trainer)
    team = models.ForeignKey(Team)
```

<https://github.com/cbedoy/CBGrandSlamDjango/blob/develop/game/models.py>

Ejemplo de controlador (Python ♥)

```
def query_sum_awards_per_player(request):
    cursor = connection.cursor()
    cursor.execute(
        'select '
        'DISTINCT '
        'firstname, '
        'lastname, '
        'age, '
        'sex, '
        'facebook, '
        'web, '
        'email, '
        '(select name from game_nationality where nationality_id= game_player.id), '
        '(select sum(amount) from game_award where player_id = game_player.id) '
        'from '
        'game_player '
        'inner join '
        'game_award '
        'where '
        'game_player.id = game_award.player_id')
    results = cursor.fetchall()
    return render_to_response('hard_queries/record_amounts.html', {"results" : results})
```

<https://github.com/cbedoy/CBGrandSlamDjango/blob/develop/game/views.py>

Ejemplo de template (HTML, JS, CSS):

```
{% extends "base.html" %}
{% block header %}Report about the history of trainers{% endblock %}
{% block content %}
<div data-role="collapsible-set" data-theme="c" data-content-theme="a">
  {%for result in results %}
    <div data-role="collapsible" data-theme="a" data-content-theme="a">
      <h3><b>{{result.7}}</b> {{result.8}}</b> trainer of <b>{{result.0}}</b> {{result.1}}</b> </h3>
      <b>{{result.7}}</b> {{result.8}}</b> has been a trainer from <b>{{result.11}}</b> to <b>
        {{result.12}}</b> ,
      he is <b>{{result.13}}</b> meters
      with <b>{{result.9}}</b> years old, his student is:
      <br>
      <b>{{result.0}}</b> {{result.1}}</b> with <b>{{result.2}}</b> years old, also he has a height of
      <b>{{result.4}}</b> and his weight is
      <b>{{result.5}}</b> kg, too he is <b>{{result.6}}</b>
    </div>
  {% endfor %}
</div>
{% endblock %}
{% block footer %}
Develop by carlos bedoy :)
{% endblock %}
```

<https://github.com/cbedoy/CBGrandSlamDjango/blob/develop/game/templates/>

Gestor de urls (Python ♥)

```
from game.views import LocationCreate

__author__ = 'administrador'

from django.conf.urls import patterns, include, url
from game import views
from django.contrib import admin
admin.autodiscover()

urlpatterns = patterns('',

    url(r'^$', views.index, name='award_list'),

    url(r'^award/list$', views.AwardList.as_view(), name='award_list'),
    url(r'^award/new$', views.AwardCreate.as_view(), name='award_new'),
    url(r'^award/edit/(?P<pk>\d+)$', views.AwardUpdate.as_view(), name='award_edit'),
    url(r'^award/delete/(?P<pk>\d+)$', views.AwardDelete.as_view(), name='award_delete'),

    url(r'^country/list$', views.CountryList.as_view(), name='country_list'),
    url(r'^country/new$', views.CountryCreate.as_view(), name='country_new'),
    url(r'^country/edit/(?P<pk>\d+)$', views.CountryUpdate.as_view(), name='country_edit'),
    url(r'^country/delete/(?P<pk>\d+)$', views.CountryDelete.as_view(), name='country_delete'),

    url(r'^game/list$', views.GameList.as_view(), name='game_list'),
    url(r'^game/new$', views.GameCreate.as_view(), name='game_new'),
    url(r'^game/edit/(?P<pk>\d+)$', views.GameUpdate.as_view(), name='game_edit'),
    url(r'^game/delete/(?P<pk>\d+)$', views.GameDelete.as_view(), name='game_delete'),

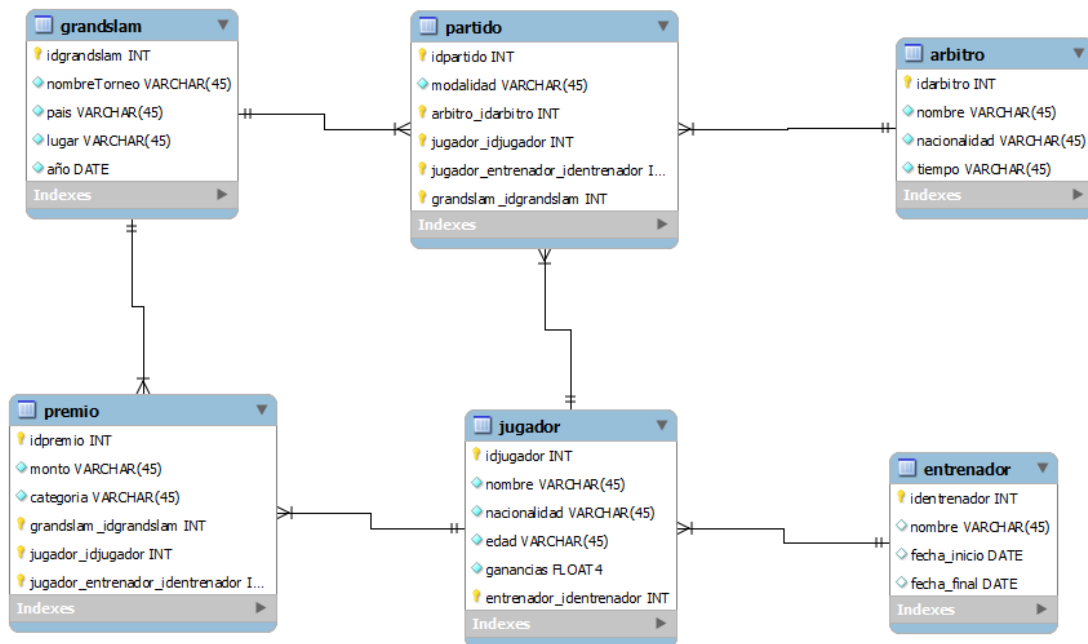
    url(r'^location/list$', views.LocationList.as_view(), name='location_list'),
    url(r'^location/new$', views.LocationCreate.as_view(), name='location_new'),
```

<https://github.com/cbedoy/CBGrandSlamDjango/blob/develop/game/urls.py>

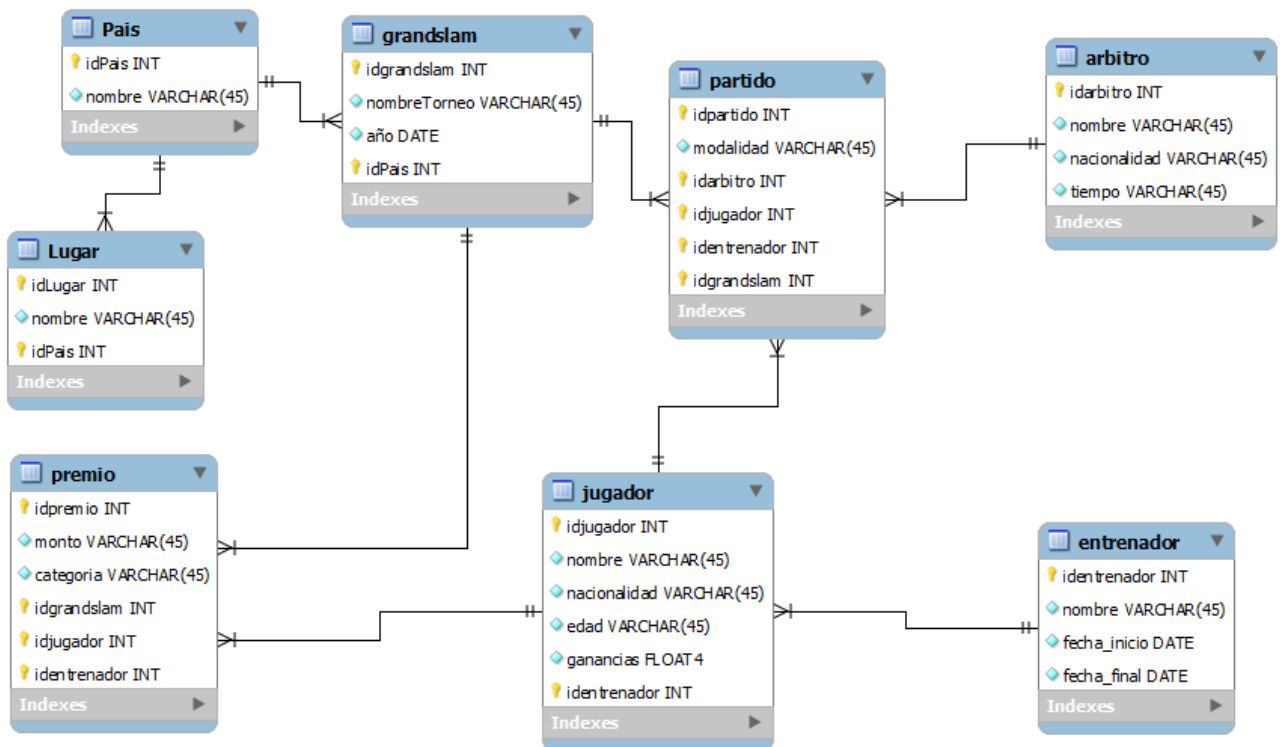
CBGRANDSLAMDJANGO

MODELO ENTIDAD RELACION

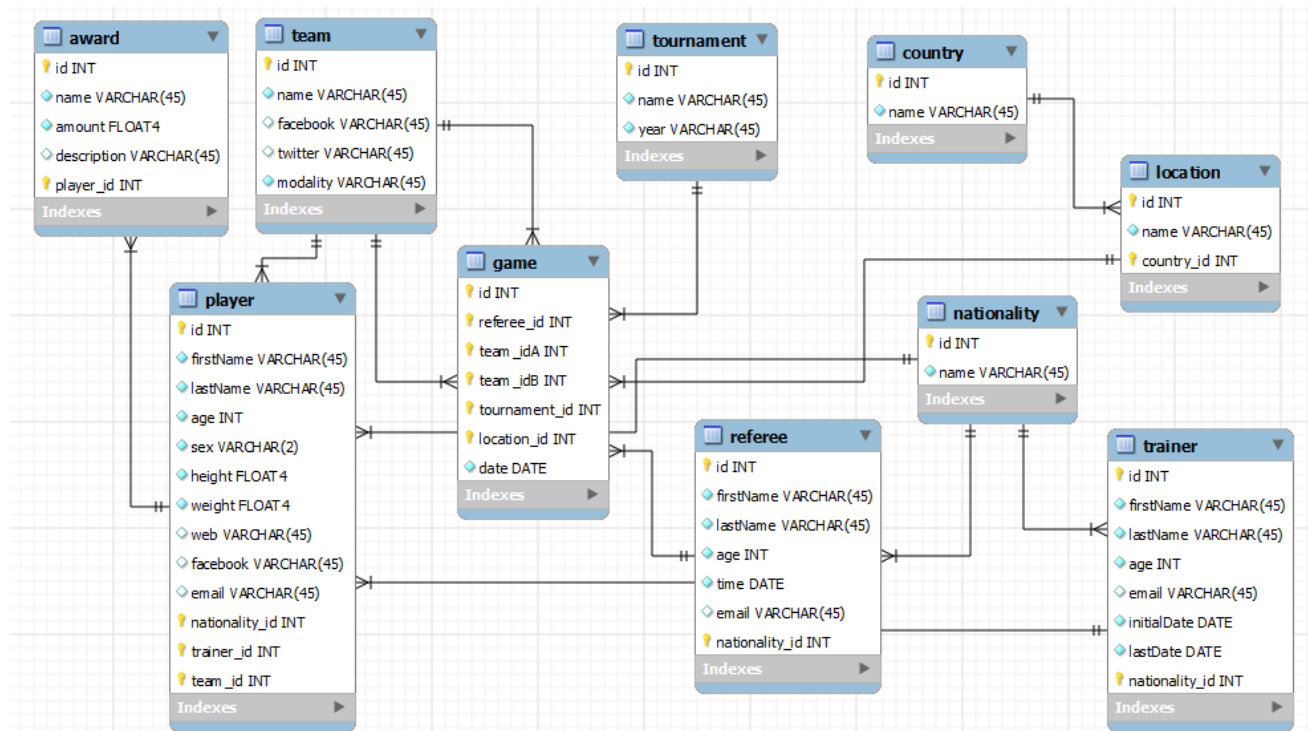
Primera conceptualización:



SEGUNDA CONCEPTUALIZACION



TERCERA CONCEPTUALIZACION NORMALIZADA



MODELO DE DATOS:

Play er												
ID	firstNa me	lastNa me	Age	Sex	Heig ht	Weig ht	Web	FB	Email	Nationality _id	Trainer_ id	Team_i d
Integer Autoincr emetal Not null Primary key	Varch ar 45 Not null	Varch ar 45 Not null	Integer not null	Varcha r 45 Not null	Integ er Not null	Integ er Not null	Varch ar 45 Not null	Varhch ar 45 Not null	Varch ar 45 Not null	Forengein key references of nationality	Forenge in key referenc es of trainer	Forenge in key referenc es of team

Trainerner								
Id	firstName	lastName	Age	Email	initialDate	lastDate	Nationality_id	
Integer Autoincremental Not null Primary key	Varchar 45 Not null	Varchar 45 Not null	Integer not null	Varchar 45 Not null	Date Not null	Date Not null	Forengein key references of nationality	

CBGRANDSLAMDJANGO

REFEREE						
Id	firstName	lastName	Age	Time	Email	Nationality_id
Integer Autoincremental Not null Primary key	Varchar 45 Not null	Varchar 45 Not null	Integer Not null	Varchar 45 not null	Varchar 45 Not null	Forengein key references of nationality

GAME						
ID	Referee_id	Team_idA	Team_idB	Tournament_id	Location_id	Date
Integer Autoincremental Not null Primary key	Forengein key of referee	Forengein key of team	Forengein key of team	Forengein key of tournament	Forengein key of location	Date Not null

AMOUNT				
ID	NAME	AMOUNT	DESCRIPTION	PLAYER_ID
Integer Autoincremental Not null Primary key	Not null Varchar 45	Not null Float	Null Varchar 45	Forengein key of player

TEAM				
ID	NAME	FACEBOOK	TWITTER	MODALITY
Integer Autoincremental Not null Primary key	Not null Varhcar 45	Null Varchar 45	Null Varchar 45	Null Varchar 45

TOURNAMENT		
ID	NAME	YEAR
Integer Autoincremental Not null Primary key	Not null Varchar 45	Integer Not null

COUNTRY	
ID	NAME
Integer Autoincremental Not null Primary key	Not null Varchar 45

CBGRANDSLAMDJANGO

LOCATION		
ID	NAME	COUNTRY_ID
Integer Autoincremental Not null Primary key	Not null Varchar 45	Not null Forengein key references of country

Reportes generados:

El sistema **CBGrandSlamDjango** será capaz manejar dentro de dos sesiones, una como backend que será el administrador del sistema quien tendrá todos los privilegios de agregar información, agregar usuarios, asignar permisos, eliminar registros, modificar registros y borrar registros, dicha vista de Logueo es la siguiente:

Django administration

Login

Username:

Password:

[Log in](#)

Así mismo continuación se muestras algunas de las vistas dentro del administrador. Nota: el sistema es responsivo esto quiere decir que la interfaz se adapta a cualquier dispositivo sea Tablet, celular o computadora.

Django administration

Welcome, adminRecent Actions

Auth

Groups

+ AddChange

Users

+ AddChange

Game

Awards

+ AddChange

Countrys

+ AddChange

Games

+ AddChange

Locations

+ AddChange

Nationalitys

+ AddChange

Players

+ AddChange

Referees

+ AddChange

Teams

+ AddChange

Tournaments

+ AddChange

Trainers

+ AddChange

Vista principal

CBGRANDSLAMDJANGO

The 'Add player' form is a web interface for adding a new player. It features a title bar 'Add player' and a yellow warning banner stating 'Fields in **bold** are required.' The form contains the following fields: 'First name:' (text input), 'Last name:' (text input), 'Age:' (dropdown menu), 'Sex:' (dropdown menu), 'Height:' (text input), 'Weight:' (text input), 'Web:' (text input), 'Facebook:' (text input), 'Email:' (text input), and 'Nationality:' (dropdown menu with a green plus icon for adding new options).

Formulario añadir jugador:

The 'Select location to change' form is a web interface for selecting a location. It features a title bar 'Select location to change' with a blue '+ Add location' button, an 'Action:' dropdown menu, a 'Go' button, and a '0 of 8 selected' indicator. The form contains a list of locations with checkboxes: 'Location', 'TOKIO', 'CHERNOBYL', 'MONDRAGON', 'CADIZ', 'AMAZONAS', 'SAN PAULA', 'QUEBEC', and 'MONTREAL'. A blue 'Save' button is located at the bottom right.

Historial de ciudades

The 'Change location' form is a web interface for changing a location. It features a title bar 'Change location' with a 'History' button. The form contains a yellow warning banner stating 'Fields in **bold** are required.' The form contains the following fields: 'Name:' (text input with 'TOKIO' entered) and 'Country:' (dropdown menu with 'JAPON' selected and a green plus icon for adding new options). At the bottom, there are four buttons: 'Delete' (red), 'Save and add another' (grey), 'Save and continue editing' (grey), and 'Save' (blue).

Opciones para modificar, eliminar o agregar otro ítem

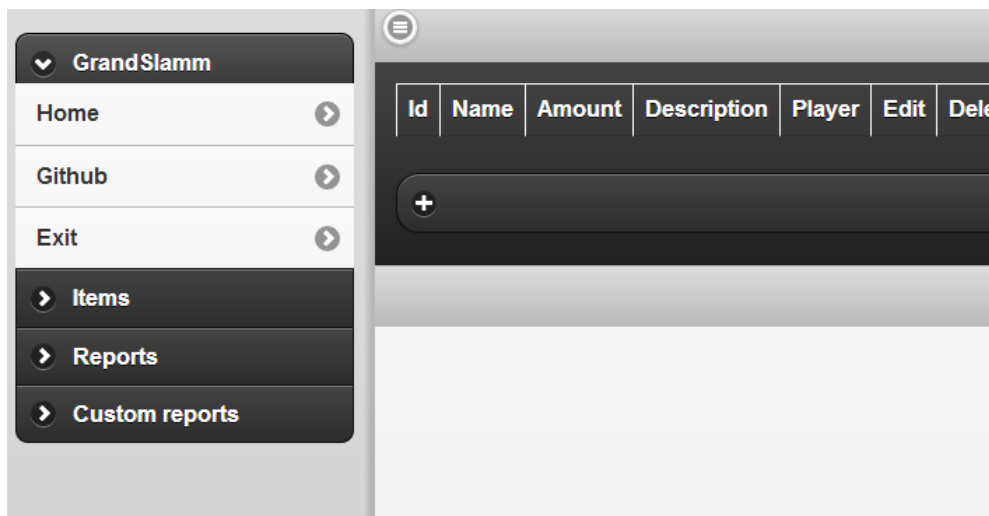
CBGRANDSLAMDJANGO

Por otra parte se tiene la aplicación que corre dentro del servidor como cliente que una aplicación totalmente enfocada a ser una aplicación móvil, dicha aplicación está desarrollada con el **framework** de **jquery-mobile** para el tratado de la vista.

La vista principal de la aplicación móvil es la siguiente:



Index.html



Menú principal de la aplicación

CBGRANDSLAMDJANGO

The screenshot shows a web application interface with a sidebar menu on the left and a main form area on the right. The sidebar menu includes 'GrandSlamm', 'Items', and 'Reports'. Under 'Items', there are links for Award, Country, Game, Location, Nationality, Player, Referee, Teams, Tournament, and Trainer. Under 'Reports', there is a link for Custom reports. The main form area is titled 'Add player' and contains fields for First name (Carlos Alfredo), Last name (Cervantes Bedoy), Age, Sex, Height, Weight, and Web (cbedoy.github.io).

GrandSlamm

Items

Award

Country

Game

Location

Nationality

Player

Referee

Teams

Tournament

Trainer

Reports

Custom reports

First name:

Carlos Alfredo

Last name:

Cervantes Bedoy

Age:

Sex:

Height:

Weight:

Web:

cbedoy.github.io

Menu para añadir items, junto a a la vista de añadir player

The screenshot shows a web application interface with a sidebar menu on the left and a main table area on the right. The sidebar menu includes 'GrandSlamm', 'Items', and 'Reports'. Under 'Reports', there are links for Award, Country, Game, Location, Nationality, Player, Referee, Teams, and Tournament. The main table area is titled 'Report referees' and contains a table with columns: ID, First name, Last name, Nationality, Age, Time, Email, Edit, and Delete. The table has 5 rows of data. Below the table is a 'New item' button.

Report referees

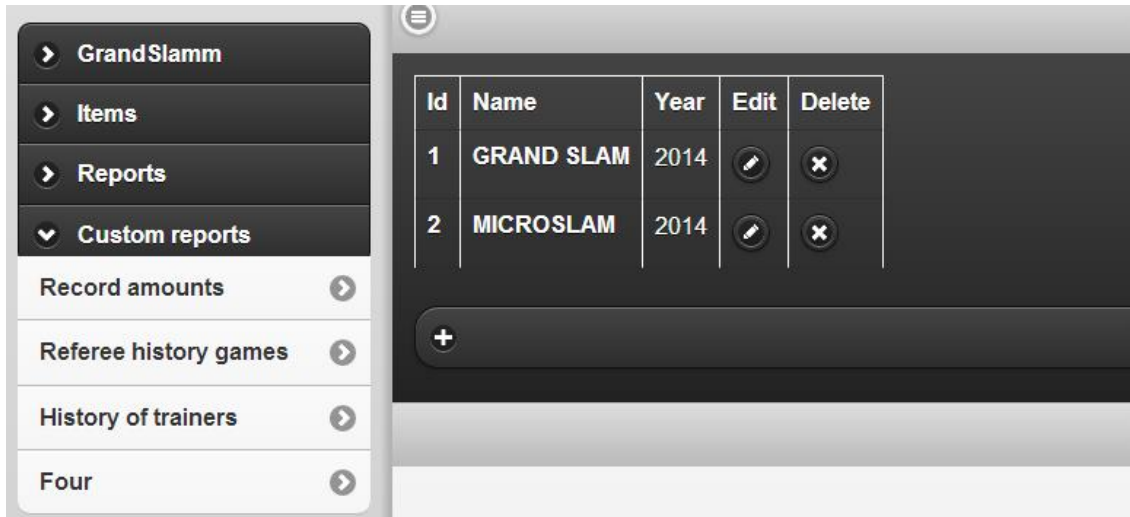
ID	First name	Last name	Nationality	Age	Time	Email	Edit	Delete
4	NOE	FLORES	SPAIN	33	Half-time	NOE@GM		
5	JUAN	SANCHEZ	SPAIN	29	All-time	JAUN@GM.COM		
6	SHIKI	TOIKI	JAPANESE	29	All-time	SHIKI@GOCM.		
7	SHAKIO	KJUNK	JAPANESE	30	All-time	SHJK@GOM		
8	PAULO	CHOELO	BRAZILIAN	30	All-time	PAUL@GOMC.		

+ New item

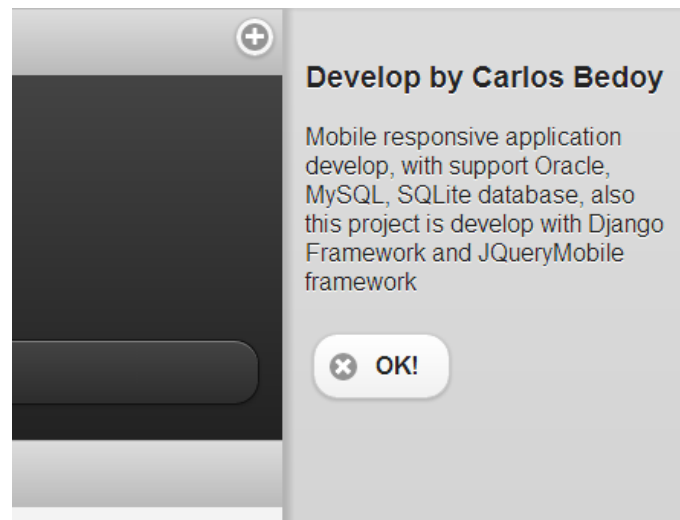
Develop by carlos bedoy :

Menu para obtener reportes junto al reporte de los referes.

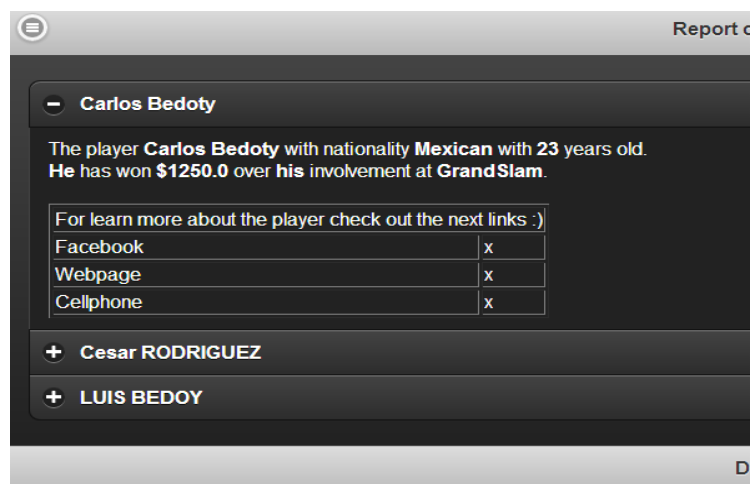
CBGRANDSLAMDJANGO



Menu para obtener los reportes customs




Creditos del desarrollador.



Algunos de los reportes generados

CBGRANDSLAMDJANGO

Report about the history referee games

+ Xabbi Toki at GRAND SLAM

+ Kowa Toma at GRAND SLAM

- Cuchi Chukig at GRAND SLAM

Cuchi Chukig has participated at the tournament **GRAND SLAM** in the date **April 14, 2014** this only have **38** years old, he appears in the **Half-time**

More information here	
Nationality	Cuba
Email	ca@gm.com


+ Stefano Ortega at MICROSLAM

+ Cuchi Chukig at MICROSLAM

+ Cuchi Chukig at Junior Slam

+ Total of games: 6

Develop by carlos bedoy :)

Report about the history of trainee

+ JesusCagide trainer of Carlos Bedoty

- LuisRangel trainer of KARlo MARIN

Luis Rangel has been a trainer from **April 1, 2014** to **April 30, 2014** , he is **Mexican** meters with **30** years old, his height is **178** cm, his weight is **80** kg, too he is **Mexican**

+ JorgeRebollo trainer of LUIS BEDOY

+ JorgeRebollo trainer of Cesar RODRIGUEZ

Develop by carlos bedoy :)

CBGRANDSLAMDJANGO

En pocas palabras el sistema permite la alta, bajas, consultas y cambios de todas las entidades así como generar reportes que se requieran atravez de comandos sql como:

- Obtener la sumatoria de premios ganados por cada usuario
- Obtener la cantidad de usuarios que han participado en X torneo y reportar cada uno de sus nombres etc.
- Obtener los jugadores que ah entrenado dicho entrenador a los largo de su vida
- Obtener los jugadores que ah referenciado dicho entrenador a los largo de su vida
- Generar practicamente cualquier reporte que proponga el maestro.

El codigo fuente del proyecto esta alojado en mi repositorio para observar cambios de codigo o bien añadir o eliminar medulos en cualquier momento del espacio

<https://github.com/cbedoy/CBGrandSlamDjango/tree/develop>

Descargar codigo fuente del proyecto

`git clone https://github.com/cbedoy/CBGrandSlamDjango.git`

`cd CBGrandSlamDjango`

`git checkout develop`

`python manage.py runserver`