

**PROYECTO DE BASES DE DATOS
B-2018**

**SALA DE JUEGOS
ONLINE**

Desarrolladores:
Charles Benitez CI: 21.185.134
Miguel Herize CI: 25.832.303
Profesora:
Solazver Sole

Objetivo General



Este proyecto tiene como objetivo el desarrollo de una **plataforma web** online en la cual los usuarios podrán elegir diferentes modalidades de juegos de su preferencia para jugar contra otros usuarios. En inicio esta plataforma web tendrá juegos de mesas estándares como Ajedrez, Póker, Ludo, Domino, entre otros.

La idea principal es crear una red social multijugador, donde los usuarios puedan loguearse en la página web y puedan acceder a diferentes eventos de diversión online, comunicarse con otros usuarios que compartan sus gustos con estos juegos de mesa, intercambiar

información, aprender, practicar, observar, ser amigo y seguir a cualquier usuario del mundo y cualquier otra necesidad disponible que los usuarios necesiten para estos divertidos y entretenidos juegos online.

URL de github: <https://github.com/migherize/charmiDjango.git>

Requisitos:

1. Debe poder loguear múltiples usuarios.
2. Modificar el perfil de los usuarios.
3. Acceder a la sala de los juegos.
4. Conectarse y cerrar sesión.
5. Modificación de los rankings.

Especificación de requisitos:

- 1.1 Cada usuario puede registrarse creando un user único sin ningún problema en la página.
- 2.1 Cuando está en la página puede modificar su perfil y terminando de llenar los campos como la dirección, foto, género.
- 3.1 Puede retar a cualquier usuario que esté conectado en diferentes juegos ya sea en poker, ajedrez u otros.
- 4.1 Conectarse y desconectarse solo desde un host no pueden haber 2 cpu usando el mismo user.
- 5.1 Cada vez que juegue contra un rival su ranking será modificado automáticamente.

Características Del producto:

- Interfaz agradable a la vista.
- Fácil entendimiento para retar.
- Página de juego de multisalas.

Metodología:

Usaremos el método ágil "Scrum" porque vamos desarrollando la aplicación de forma interactiva y al final es que haremos las pruebas de las funciones. Las funciones y el diseño de la aplicación fue lo primero que planteamos por esto usamos dicha metodología.

Estilo arquitectónico:

El estilo arquitectónico que creemos apropiado es el MVC, (modelo-vista-controlador). Donde tenemos:

- El componente Modelo que gestiona los datos del sistema como son las clases (modelos) de nuestro proyecto, en las cuales estarán todos los datos (Base de datos), atributos y variables hechas en el lenguaje de programación python.
- El componente Vista, la cual es la capa de la lógica del proyecto. Esta capa contiene la lógica que accede al modelo y a la plantilla apropiada: se puede pensar como un puente entre los modelos el modelo y las plantillas.

-El componente controlador el cual se encarga de la comunicación entre 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 el controlador es manejada por el mismo framework, Django es conocido como un framework MTV (modelo-template-vista). De donde podemos decir que la T es la capa de presentación. Esta capa contiene decisiones relacionadas a la presentación: como algunas cosas son mostradas sobre una página web o otro tipo de documento.

Implementación de funciones almacenadas o triggers

La utilidad principal de un trigger es mejorar la gestión de la base de datos, ya que no requieren que un usuario los ejecute.

Una parte muy importante de nuestro proyecto es la actualización de los puntos de ranking de cada usuario justo después de que termine cada partida, por lo tanto, un trigger a usar sería **UPDATE** con mucha frecuencia ya que siempre va a estar actualizando la puntuación de cada usuario

INSERT También será utilizado para expandir el universo de nuestra sala de juego cuando un jugador nuevo se registre con su Nickname.

Un trigger o disparador son empleados para implementar las REGLAS DE NEGOCIO (tipo especial de integridad) de una base de datos. Una Regla de Negocio es cualquier restricción, requerimiento, necesidad o actividad especial que debe ser verificada al momento de intentar agregar.

Podemos aplicar restricciones cuando un usuario desee filtrar partidas por niveles de juego. También verificar cuando un usuario nuevo se registre ya que no pueden haber más de dos nickname iguales.

Tecnologías que usaremos para desarrollar el proyecto:

- Como lenguaje de programación Python3
- Como gestor de base de datos se usará PostgreSQL
- Como Framework utilizaremos Django
- Como lenguaje etiquetado HTML5, CSS, JavaScript.

Transacciones:

En nuestro proyecto podemos notar transacciones cuando el usuario quiera actualizar su perfil.

Cuando un jugador termina una partida se cambiará su puntuación y su número de victorias, derrotas o empate.

Cuando un usuario quiera restringir el nivel de los jugadores que desee enfrentar.

Veámoslo en el siguiente ejemplo:

x = ranking usuario1
y = ranking usuario2
n = número fijo de perdida.

Supongamos que gane el Usuario 1 una transacción válida con un plan de ejecución.

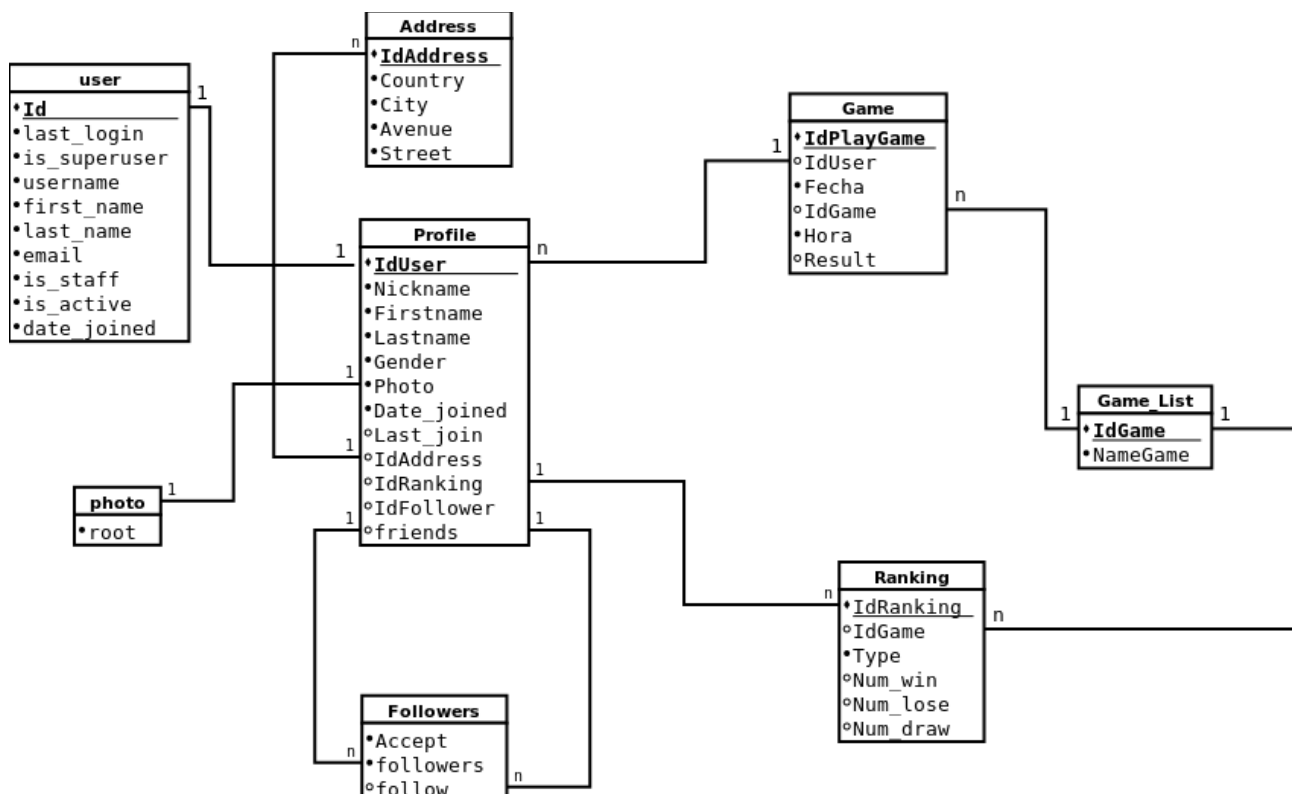
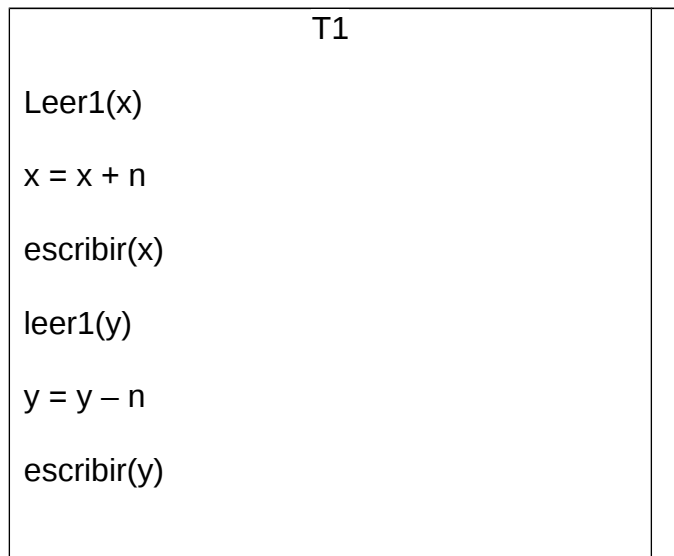
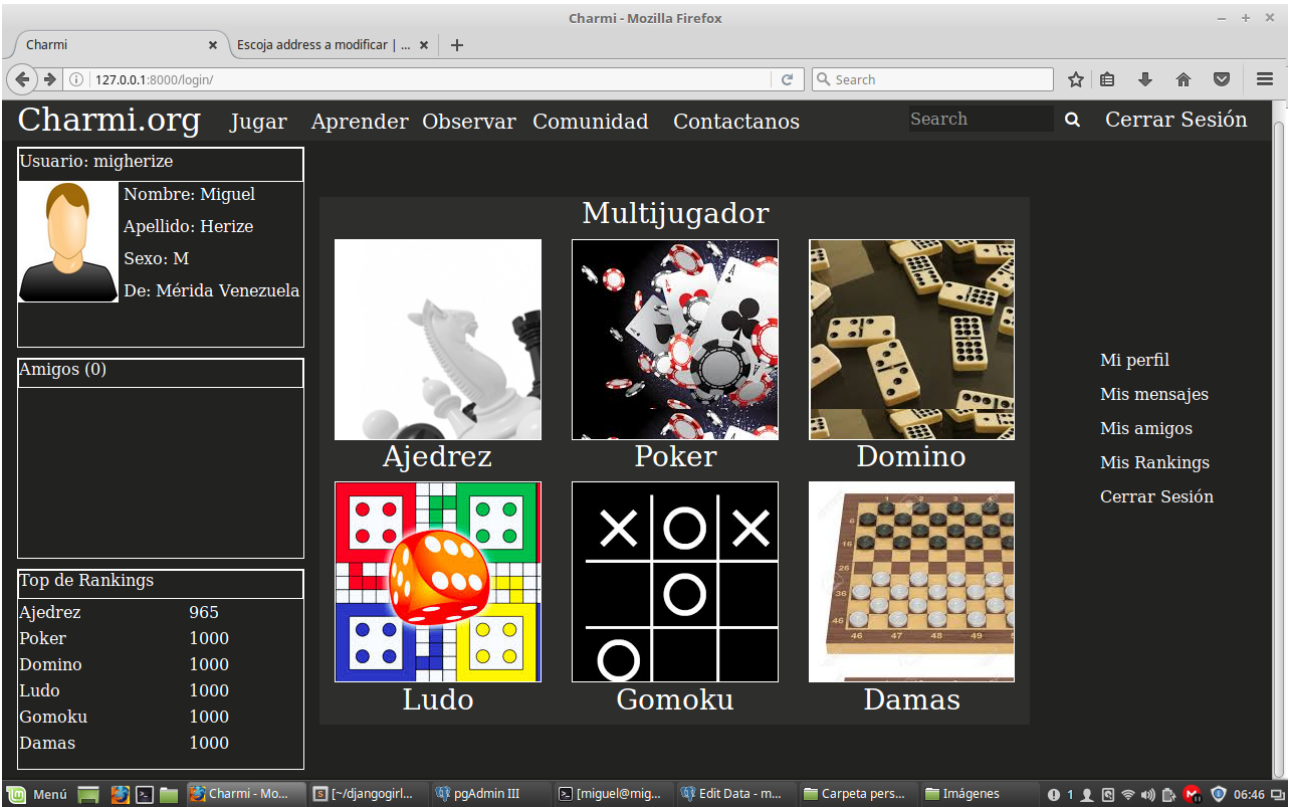
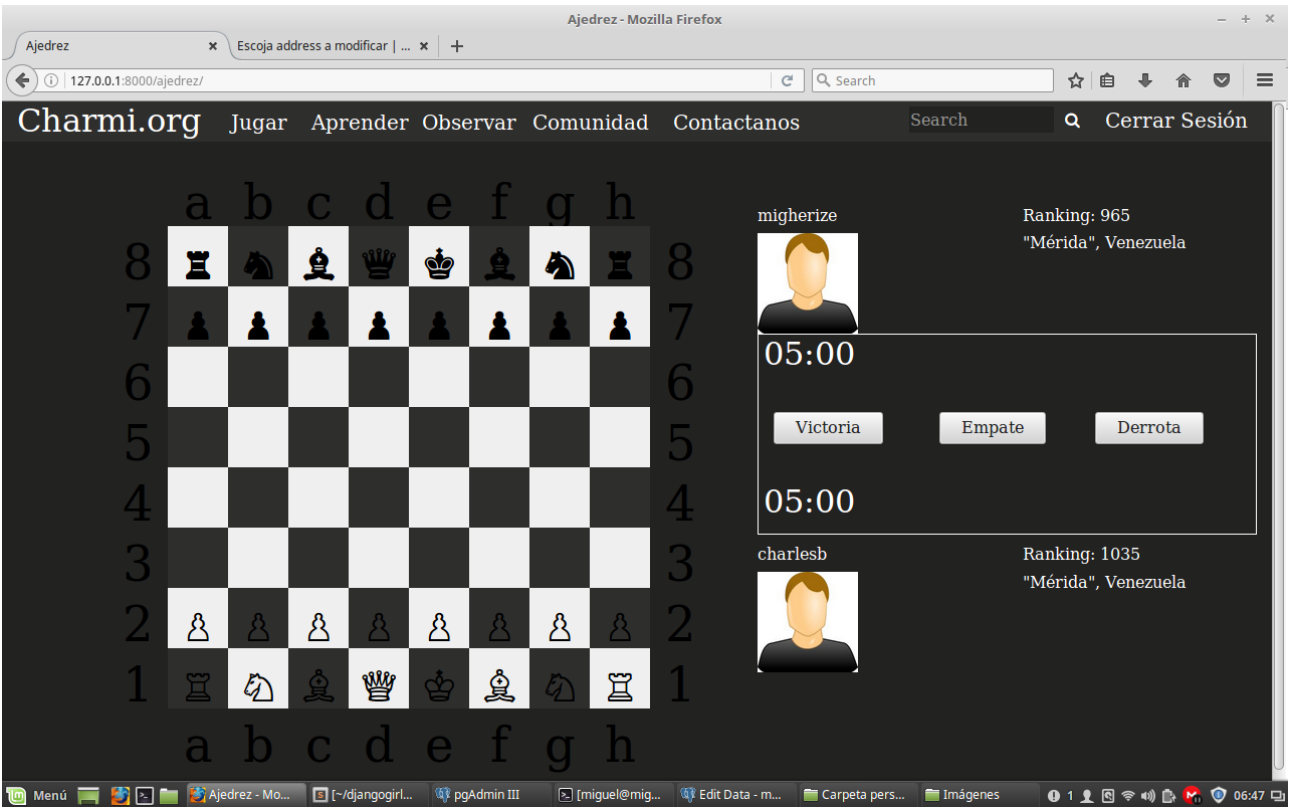


figura1. Diagrama de relación de la base de datos

Mockups:



Sala principal



Sala de juego