

# Proyecto Final de Modelos de Optimización II ESTADÍSTICAS PARA CONOCER AL AÑO



UNIVERSIDAD DE LA HABANA  
FACULTAD DE MATEMÁTICA Y  
COMPUTACIÓN

Integrantes Equipo 6:  
Olivia González Peña C-411  
Sheyla Cruz Castro C-412  
Laura Brito Guerrero C-412  
Juan Carlos Casteleiro Wong C-411

## Índice general

<b>Problema</b> .....	2
<b>Solución</b> .....	3
2.1. Implementación del Back-End .....	3
2.2. Implementación de la API.....	5
2.3. Front-End.....	9
2.4. Cómo poner a funcionar la aplicación .....	14
<b>Anexos</b> .....	16

---

# 1. Problema

## Planteamiento del problema

**Usuario final:** Fernando Rodríguez Flores @fernan2rodriguez

## Objetivos

El objetivo de este proyecto es diseñar e implementar un sistema (¿de encuestas? ¿de preguntas? ¿de tarjetas por el día de las madres?) que permita obtener información sobre un grupo de clases al comienzo del curso. Entre lo que se desea conocer están los intereses de los estudiantes (qué les gusta y a qué le dedican su tiempo libre). Cuánto conocen sobre determinados temas (videojuegos, películas, libros). Si juegan, leen o consumen series, y en esos casos qué tipos de juegos, libros, o series. Por ejemplo, en caso de que a alguien le guste la poesía, qué tipo de poesía, qué autores. ¿Qué tipo de música? Etc. ¿Cuál es su relación con las asignaturas que ya vieron? ¿Cuáles le gustaron? ¿Cuáles no les gustaron? ¿Cuáles les dio absolutamente lo mismo? ... etc.

El objetivo de esta primera parte es determinar qué tipo de elementos y referencias se pueden incluir en las conferencias y clases prácticas de forma que sean relevantes para la mayor cantidad de personas posibles. También interesa conocer los niveles de creatividad y de sentido del humor que tengan los estudiantes. Esto es a través de los *tests* y *cuestionarios* ya establecidos para esos fines.

Algunas de estas informaciones conviene que sea *con nombre*, otras a lo mejor conviene que sea anónima.

Como parte del proyecto de debe:

1. Crear un mecanismo que permita recoger esta información *de la mejor manera posible*. *De la mejor manera posible* significa que quizás la mejor opción no sea ponerles una encuesta con varios cientos de incisos ... o a lo mejor sí.
2. Crear un sistema que permita analizar la información recogida.
3. Mostrar la información recopilada de una manera que resulte útil para la planificación de las asignaturas.

---

## 2. Solución

**2.1 Implementación del Back-End** La solución que concebimos para nuestro problema, requiere la implementación de diversas estructuras de datos que nos permitan manejar la información más cómodamente. La lógica que seguimos para el algoritmo está basada en el análisis estadístico de las respuestas de los estudiantes a las encuestas, tanto de manera individual como general. A continuación, los detalles:

- **poll.py**: En este fichero se recogen los datos de la aplicación, es decir, a los estudiantes, los temas a considerar en las encuestas y las respuestas de los estudiantes a las mismas. Dicha información se recoge en forma de diccionario. Se tiene en consideración que las encuestas pueden ser tanto anónimas o no, esto no afecta a los algoritmos diseñados.

En el método **analyze** se recoge esta información y conforme a esto se crean las estructuras de datos correspondientes, primeramente se diseña un contexto de temas (**Theme\_Context**), el cual ordena eficientemente los temas siendo estos a la vez subtemas de otros temas, esto se diseña jerárquicamente.

A la vez, los estudiantes se recopilan en **Student\_Preferences**, donde se guardan los datos de los estudiantes en caso preciso y las respuestas de los mismos a las encuestas.

Se analizan los estudiantes en pares no repetidos, se tiene la certeza que si se compara el estudiante  $i$  con el estudiante  $j$  se obtienen los mismos resultados que si se analiza al estudiante  $j$  con el  $i$ . Se comparan los criterios de preferencia de los estudiantes, qué tan alejados o cercanos están los mismos en sus gustos. Antes de explicar el proceso de comparación se pasa a explicar la organización de las preferencias de cada estudiante.

Una vez que se recojan las respuestas de cada estudiante como (*Student\_Preferences*) en **answers** (diccionario con llave *tema* y valor correspondiente: *respuesta del estudiante en ese tema*), se pasa a analizar sus preferencias (**preferences**) en orden descendente.

Se implementa una clase **AVLSearchBinaryTree\_Insert** la cual crea un AVL con valores en los nodos (*tema, cantidad de respuestas al tema*), mientras mayor cantidad de aciertos tenga ese tema (que se considera una hoja de la jerarquía de los temas, ya que los temas *padres* se van desglosando en subtemas *hijos*, los cuales a su vez tie-

---

nen su propia jerarquía, hasta caer en un subtema *hoja*, en la cual los estudiantes desarrollan o *marcan* sus gustos correspondientes), mayor es su preferencia. Estos nodos se organizan por el segundo valor de la tupla en forma ascendente. Luego de estar confeccionado el AVL, se realiza un recorrido entre-orden y devuelve una lista ordenada de las preferencias y las mismas se guardan en orden inverso en **preferences**.

Como por cada estudiante se tiene un orden en sus preferencias, se analizan los pares de estudiantes, esto se implementa en **invert\_count**. Se toman **s1.preferences** y **s2.preferences** (**s1** y **s2** estudiantes) y se comparan, donde sin pérdida de generalidad se considera como arreglo ordenado a **s1.preferences**. Entonces, se ordena **s2.preferences** respecto a **s1.preferences**, y dependiendo de la cantidad de inversiones que tenga  $tema_{i2}$  (el tema  $i$  de **s2**) con respecto a  $tema_{i1}$  (tema  $i$  de **s1**) se establece la lejanía o cercanía del  $tema_i$  que presentan ambos estudiantes. El criterio de cercanía (**s.likes**) o lejanía (**s.dislikes**), lo brinda un porcentaje, si la cantidad de inversiones es menor que un  $p$  por ciento (no se define porque cada tema o encuesta tiene su propio patrón definido) entonces el  $tema_i$  es común entre ambos estudiantes en su orden de preferencias. En caso contrario, los gustos de ambos estudiantes con respecto al tema están alejados. Después de conocer la relación entre ambos estudiantes se añade a las estructuras de datos correspondientes en la instancia de **s1** y **s2**.

Luego de recorrer y analizar los gustos de todos los pares de estudiantes, se recurre al análisis general estadístico, el cual se guarda en la instancia de **Theme\_Context**, y se implementa en el método **theme\_context.stadistics\_result**.

- **hierarchy.py** : En dicho fichero se implementa la clase **Theme\_Context** y **Theme\_Scope**. En **Theme\_Scope** se encuentra la jerarquía de los temas, y en **Theme\_Context** se guardan estos temas y se realizan los análisis referentes a los mismos, como por ejemplo el análisis estadístico con respecto a las respuestas de los estudiantes.
- **likes\_dislikes.py**: Se encuentra la clase **Student\_Preferences**, donde se guardan los datos de cada estudiante, sus respuestas a las encuestas, la relación que guarda este con los  $n - 1$  estudiantes restantes, sus preferencias con respecto a los temas.
- **invert\_array.py**: Se realiza el análisis **invert\_count** mencionado anteriormente.
- **stadistics.py**: Se implementa la clase **Stadistics\_Op**, la cual se encarga de realizar el análisis estadístico por tema. Se implementa el

---

cálculo de las medidas de Tendencia Central y las medidas de Dispersión, necesarias para una correcta interpretación. Dicho análisis por tema se instancia en 2.1, donde se guarda en el diccionario **stadistics** (*tema : análisis estadístico*).

**2.2 Implementación de la API** Para implementar un sistema capaz de procesar y mostrar los resultados de una manera atractiva para el usuario final, decidimos desarrollar una aplicación web que consta de dos partes, una aplicación visual desarrollada en ReactJS y otra aplicación tipo cliente-servidor conocida como interfaz de programación de aplicaciones(API) desarrollada en Django.

En principio nos decidimos por el framework Django por ser de los mejores del mercado para el lenguaje Python, uno de los lenguajes más utilizados hoy en día tanto en el desarrollo web, como en la inteligencia artificial y el Big Data, y en el cual desarrollamos todo el backend del proyecto, puesto que es un lenguaje que destaca por ser de código abierto, multiplataforma y muy simple y fácil de entender.

Django es lo que se conoce como un framework “full stack” para Python con el cual se pueden abordar todo tipo de proyectos como el desarrollo web escalable y de alta calidad, por lo que nos fue sencilla la decisión de eventurarnos al uso de este nuevo framework con el que no habíamos trabajado antes. Django destaca por sus múltiples ventajas como son gran rendimiento y flexibilidad; trabaja bajo el patrón MVC (Modelo Vista Controlador), ya conocido por nosotros, lo que permite un desarrollo ágil y reutilizable; incorpora una amplia variedad de paquetes de librerías, manejo de versiones que permite una distribución simple de actualizaciones; cuenta con panel de administración para bases de datos, consta de un mapeador objeto-relacional, un sistema de autenticación de usuarios, brinda vistas genéricas y un sistema extensible de plantillas basado en etiquetas, urls con soporte para expresiones regulares; en fin, todo esto justifica la inmensa comunidad de usuarios de que dispone en internet [6].

Para modelar el problema y manejar los datos de manera persistente se diseñó un modelo de cuestionarios con preguntas y respuestas como entidades relacionadas entre sí, además de otra tabla de temas y subtemas asociados a dichas preguntas, para esto utilizamos la base de datos relacional **postgresql** y el gestor de base de datos **pgadmin**. Este modelo de cuestionario más abierto se usó para modelar las encuestas de

---

tipo tiempo libre, por otro lado para gestionar la información los tests de creatividad y humor, y el nivel de agrado de las asignaturas se crearon igual entidades independientes registradas en base de datos(Ver Figura [7]).

Como es común en Django para cada funcionalidad crear aplicaciones que funcionan como módulos completamente independiente, fraccionamos nuestra API en tres aplicaciones, una de ellas es **questionnaire**, que contiene los siguientes modelos:

- **Questionnaire** con dos propiedades, una *name* con el nombre del cuestionario y otra *pub\_date* con la fecha de creación del cuestionario.
- **Themes** con un campo *name* con el nombre del tema, y otro *father* con el nombre del tema padre al que está asociado, de tener el campo padre ocupado se convierte automáticamente en subtema, de lo contrario se considera simplemente un tema.
- **Question** tiene un campo *text* con la pregunta en cuestión, y dos llaves foráneas con el cuestionario y el tema o subtema al que pertenece, además de dos propiedades para ver la lista de opciones y respuestas asociadas a la pregunta.
- **Answer** con un campo *answer* con la respuesta de la pregunta, y dos llaves foráneas, una a la pregunta y otra al usuario que esta respondiendo, este modelo usuario lo discutiremos más adelante al igual que como hacemos para gestionar el anonimato de los encuestados.
- **Option** con un *value* con el valor de la opción y una llave foránea a la pregunta de la cual es opción.

Para los tests de creatividad y humor, y para gestionar la información de las asignaturas creamos otra aplicación llamada **averagetest** con los siguientes modelos:

- **AverageTest** este modelo contiene dos campos, uno llamado *category* para distinguir entre un test de humor y uno de creatividad, y otro *value* para guardar el resultado del test, y una llave foránea asociada al usuario que realizó el test.
- **SubjectTest** contiene un campo *name* con el nombre de la asignatura y otros tres campos *likes*, *dislikes* y *dontcare* con la cantidad de me gusta, no me gusta y me da igual de dicha asignatura.

Tenemos además los siguientes endpoints de los que consume la aplicación visual para que el usuario interactúe con el sistema y conozca los datos que le interesan de una manera atractiva:

- 
- ✓ <http://127.0.0.1:8000/questionnaire-create/> para crear un cuestionario de tipo tiempo libre.
  - ✓ <http://127.0.0.1:8000/questionnaire-response/> para guardar una respuesta de un cuestionario de tipo tiempo libre.
  - ✓ <http://127.0.0.1:8000/questionnaire-process/> para procesar los resultados de las encuestas de tipo tiempo libre, aquí se recopilan todas las encuestas realizadas y el conjunto de temas y subtemas registrados en el sistema, dicha información se le facilita al backend para su posterior análisis, estos resultados son guardados en files que se generan dentro de la API en un espacio destinado para ello.
  - ✓ <http://127.0.0.1:8000/process-most-popular-themes/> para procesar de acuerdo a la cantidad de votos los temas más populares, aquí se hace un análisis a partir de todas las encuestas realizadas y por cada selección de un subtema específico se contabiliza la cantidad de selecciones del tema padre que contiene dicho subtema, luego estos resultados se ordenan de manera descendente, quedando en el top los más votados, por ende los más populares y son devueltos a la aplicación visual en formato JSON para que sean mostrados al usuario final.
  - ✓ <http://127.0.0.1:8000/process-most-popular-subthemes/> para procesar de acuerdo a la cantidad de votos los subtemas más populares, aquí se hace un análisis similar al explicado anteriormente, solo que en este caso contabilizamos los subtemas de manera independiente, luego estos resultados por subtemas se ordenan de manera descendente, quedando en el top los más votados, por ende los más populares y son devueltos a la aplicación visual en formato JSON para que sean mostrados al usuario final.
  - ✓ <http://127.0.0.1:8000/process-average-test/creativity/> o <http://127.0.0.1:8000/process-average-test/humor/> para guardar una respuesta del test de creatividad o de humor según el tipo especificado en la url.
  - ✓ <http://127.0.0.1:8000/process-years-average/humor/> o <http://127.0.0.1:8000/process-years-average/creativity/> para procesar los resultados de los test de creatividad o de humor según el tipo especificado en la url, aquí se recopilan todos los resultados del test en cuestión y se retorna en formato JSON el promedio de los datos para que sean mostrados al usuario final.
  - ✓ <http://127.0.0.1:8000/process-years-particularities/humor/> o <http://127.0.0.1:8000/process-years-particularities/creativity/> para procesar los resultados de los test de creatividad o de humor



---

según el tipo especificado en la url, aquí se recopilan todos los resultados del test en cuestión y se contabilizan y clasifican en las distintas categorías que serán descritas posteriormente, luego son devueltos a la aplicación en formato JSON las puntuaciones obtenidas en cada categoría.

- ✓ <http://127.0.0.1:8000/process-subject-test/> para guardar una respuesta del test de asignaturas, donde el usuario decide se le gustan, le disgustan o le da igual cada una de las asignaturas del test.
- ✓ <http://127.0.0.1:8000/process-subject-test-result/Álgebra/> para procesar los resultados del test de asignaturas para una asignatura específica, aquí se recopilan todos los resultados obtenidos en esa asignatura y se retorna a la aplicación visual en formato JSON la cantidad de likes, dislikes y dontcares para que sean mostrados al usuario final.
- ✓ <http://127.0.0.1:8000/process-subject-test-all-result/> para procesar los resultados del test de asignaturas de todas las asignaturas, aquí se recopilan todos los resultados obtenidos en cada una de las asignaturas y se retorna a la aplicación visual en formato JSON la cantidad de likes, dislikes y dontcares de todas estas para que sean mostrados al usuario final.

Por otro lado trabajamos en una aplicación llamada **user** con el objetivo de crear un proyecto más robusto y extensible para su posterior desarrollo, aquí creamos un modelo usuario con un conjunto de campos como *name*, *last\_name*, *email* y otras propiedades para controlar los permisos del usuario, códigos de verificación para temas de autenticación, además trabajamos en una tabla **role** para gestionar los roles de los usuarios y permitir que la aplicación cuente con roles como estudiante, profesor y administrador. Se implementaron además endpoints para el log-in y registro de usuarios a la aplicación, cambio y recuperación de contraseña, y para el tema de la autorización se trabajó con JWT.

Esta aplicación está implementada y pendiente a mejoras, no obstante se trataron de anónimas las encuestas por tanto todos los users creados en la aplicación son falsos, estos se utilizan solamente para controlar el número de encuestados y distinguir entre las respuestas que se brindan por diferentes encuestados. Para el acceso a la API se trabajó con *superusers* que se crearon a nivel de consola y se registraron en la aplicación, capaces de navegar por la API y gestionarla a su totalidad, pues el framework incluye vistas genéricas para trabajar con los modelos, lo que permite crear, editar, borrar y trabajar de manera general con los modelos antes mencionados.

---

**2.3 Front-End** Con la intención de crear vistas interactivas y de fácil manejo tanto para los usuarios como para los desarrolladores que en el futuro se encarguen del mantenimiento y actualización del proyecto, utilizamos la biblioteca ReactJS. ReactJS es una plataforma de código abierto que brinda acceso para usar bibliotecas y marcos de forma gratuita. Al estar basado en JavaScript, cuenta con una amplia comunidad que dispone de un gran número de bibliotecas externas. La principal ventaja de React es poder generar el DOM (Modelo de Objetos del Documento, estructura de los elementos que se generan en el navegador web al cargar una página) de forma dinámica. Esto permite que para poder visualizar los cambios de los datos, no es necesario renderizar toda la página de nuevo, sino solamente el componente que haya sido actualizado. Gracias a esta característica mejora: la experiencia de usuario al navegar por la aplicación web, la rapidez en la carga de las páginas y facilita el mantenimiento de la aplicación. Las aplicaciones webs desarrolladas con React están basadas en componentes reutilizables. Esto facilita que la aplicación sea más escalable y fácil de mantener ya que los errores sucederán en la propia funcionalidad del componente o en la comunicación con los demás.

Para la creación de los cuestionarios que conforman las encuestas, hicimos uso de la biblioteca SurveyJS. Es una herramienta de código abierto que proporciona los instrumentos necesarios para crear encuestas, formularios y cuestionarios totalmente personalizables y que se pueden integrar de manera sencilla. Además, brinda la ventaja de poder previsualizar los objetos de la encuesta que se está creando, así como el código de los resultados y el formato de estos, para así establecer las pautas para la comunicación con el resto de las capas de la aplicación que lo requieren para el correcto entendimiento y procesamiento de los datos.

La aplicación recibirá a los usuarios con la vista que muestra los cuatro test que se pueden realizar, a los cuales se podrá acceder con el botón “Empezar” correspondiente a cada uno.

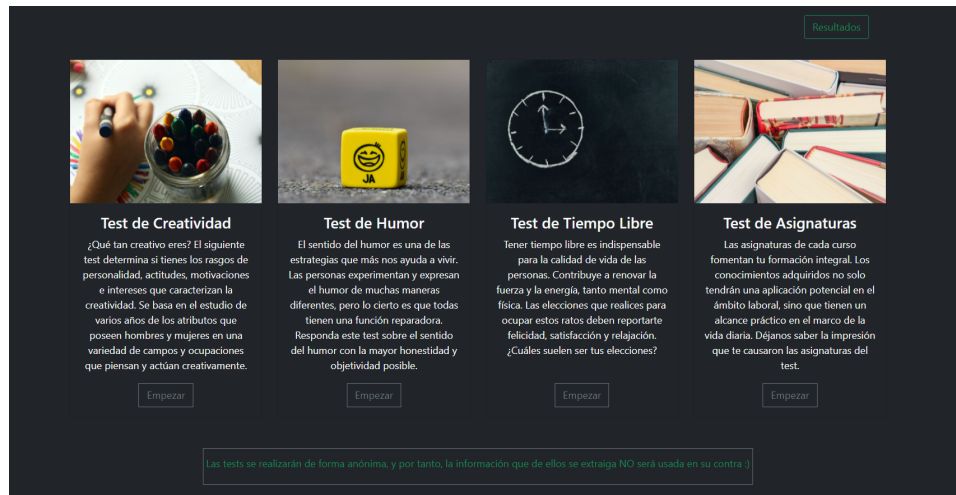


Figura 1: Página de Inicio

- **Test de Creatividad:** Para la formulación de este test nos basamos en el cuestionario [1], tanto para las preguntas como para la puntuación que se obtiene al responder cada una y la clasificación que se alcanza en cuanto a la creatividad, teniendo en cuenta la suma de los puntos acumulados por cada respuesta. A continuación, se especifican:
  - **95-116:** Creatividad excepcional
  - **65-94:** Buen índice de creatividad
  - **40-64:** Ligeramente mejor que el promedio
  - **20-39:** Creatividad promedio
  - **10-19:** Creatividad pobre
  - **< 10:** Nada de creatividad
- **Test de Humor:** Para la realización de esta encuesta, tomamos como guía las preguntas y clave de calificación del test [2], respetando las categorías en que se clasifica un encuestado según la suma obtenida por sus respuestas, que son las siguientes:
  - **115-140:** Excelente sentido del humor y actitud
  - **85-114:** Sentido del humor promedio y saludable
  - **50-84:** Sentido del humor un poco deteriorado
  - **25-49:** Deficiencia del sentido del humor
  - **< 24:** Sufre de seriositis crónica
- **Test de Tiempo Libre:** Esta encuesta fue realizada completamente por nosotros. Debido a que nuestra edad es bastante próxima a la de los futuros encuestados, tuvimos en cuenta nuestros gustos y los de nuestros compañeros y amigos. No obstante, intentando ser lo

---

más inclusivos posible y hacer que la encuesta no deje de recoger respuestas de todo tipo, introdujimos dentro de las opciones de marcado una casilla “¿Otro? ¿Cuál?” (Ver Figura [9]) que despliega un text-box para escribir una opción en caso de no encontrarse entre las que proponemos. Recurrimos a sitios que listan, por ejemplo, las mejores películas de todos los tiempos (recurrimos a [3] al haber trabajado recientemente con este sitio) o los artistas más escuchados en Spotify [4], para las selecciones más específicas.

Es necesario aclarar que todas las preguntas de la encuesta no brindan resultados del mismo tipo. Hay preguntas más genéricas cuyas respuestas pueden derivar en más preguntas; por ejemplo, la pregunta 3 de la encuesta se refiere al tipo de contenido que usualmente consumimos cuando vemos la televisión, y una de las respuestas a esta pregunta es ‘Películas’. En la pregunta 4, buscamos ser más específicos para saber de aquellos que han marcado esta opción, qué géneros prefieren, o aun más específico, si alguna de las películas que listamos en la pregunta 7 está entre sus preferidas. Contemplamos, además, el caso en el que un encuestado no suele disfrutar de la televisión y, por tanto, no tiene películas preferidas, para ello dejamos una opción de marcado ‘Ninguno’ en todas las preguntas.

Las preguntas de ámbito general constituyen los temas, y las más específicas, los subtemas a los que nos referimos anteriormente.

- **Test de Asignaturas:** Para la confección de esta encuesta, incluimos las asignaturas que se imparten en primer año de la carrera Ciencias de la Computación en el nuevo plan de estudios E.

De la página principal, accionando el botón Resultados se podrá navegar hasta la página que muestra los resultados que se han obtenido al procesar las respuestas de los encuestados de cada test.

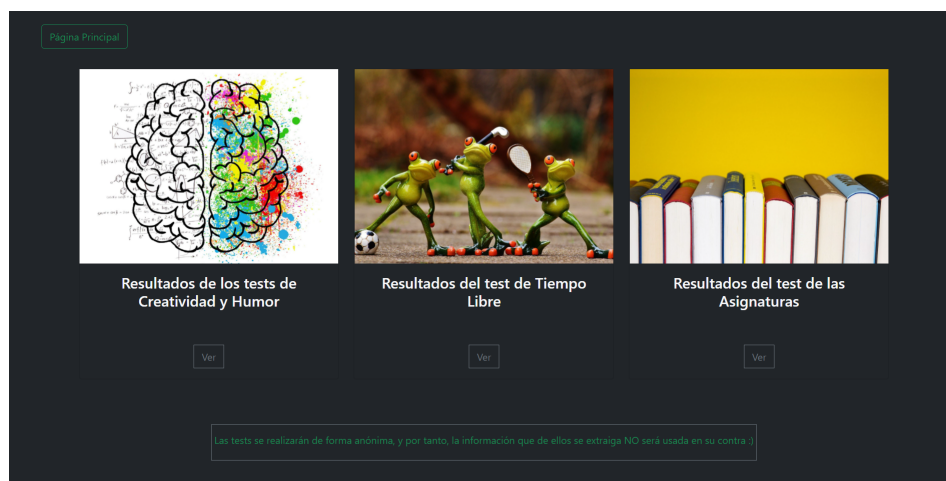


Figura 2: Página de Resultados

Los resultados de los test de Creatividad y Humor se muestran a través de gráficos de doughnuts. Cada color, y la sección correspondiente a este en el gráfico, mostrará la relación de encuestados cuyo índice de creatividad o humor concuerde con la categoría que el color en cuestión indica, según las puntuaciones. La relación entre colores y categorías se especifica al lado de cada gráfico.

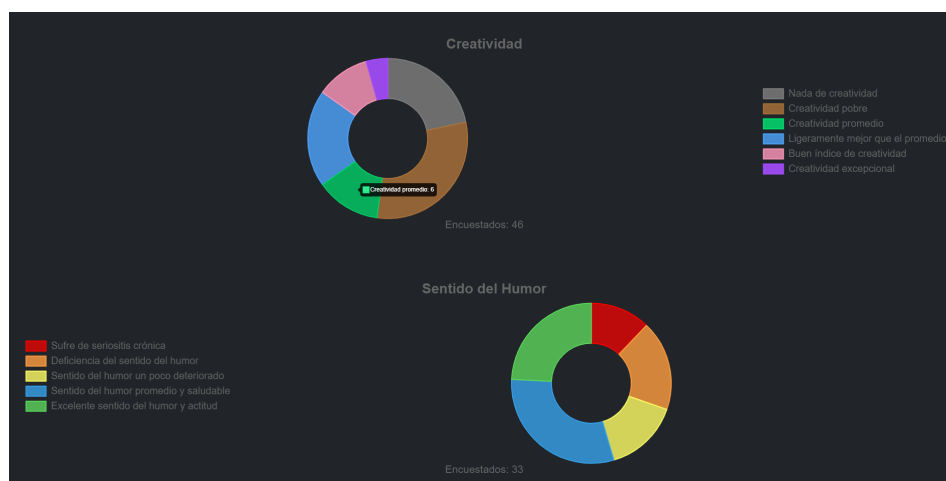


Figura 3: Resultados de Humor y Creatividad

Podrán eliminarse del gráfico, por un momento, las categorías en caso de querer, por ejemplo, comparar la cantidad de encuestados de dos categorías particulares.

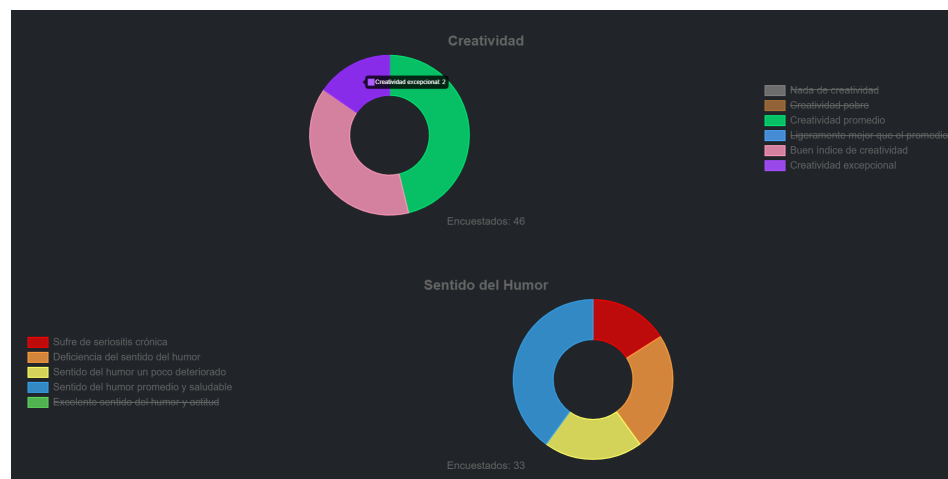


Figura 4: Resultados de Humor y Creatividad filtrados

Los resultados del test de Tiempo Libre se mostrarán en forma de ranking, es decir, una tabla donde las primeras filas son las más significativas según el criterio de ordenación escogido. Decidimos ilustrar dos tablas para mostrar una comparación entre temas y, por separado, una comparación entre subtemas. En ambos casos se compara teniendo en cuenta la cantidad de encuestados que marcaron esta opción como su preferencia.

En el ejemplo se ilustra que el tema más gustado es el **‘Deporte’**, sin embargo, dentro de este, existen dos subtemas votados: **‘Fútbol’** y **‘Tenis’**, donde el **‘Tenis’** se encuentra entre los subtemas menos votados. Esto muestra que no tiene por qué coincidir la posición de un tema con los subtemas correspondientes a él y, por ello, decidimos que se puedan observar los resultados de manera independiente. De esta forma, además, se podrá elegir entre incluir en las clases contenido relacionado con un tema general, aunque las decisiones dentro de este sean variadas y heterogéneas, que le parezca interesante o represente el agrado más común entre los estudiantes; o incluir contenido de un subtema popular dentro de un tema específico, aunque no sea el único votado dentro del tema al

Resultados

Ranking de Temas más gustados

Tema	Cantidad de Votos
Practicar Deportes	10
Películas Favoritas	8
Series Favoritas	6
Artistas Favoritos	4
Autores Favoritos	2

Ranking de Subtemas más gustados

Subtema	Cantidad de Votos
Fútbol	8
Game of Thrones	6
Titanic	5
Forrest Gump	3
Tenis	2
Coldplay	2
Lady Gaga	2

Figura 5: Resultados de Tiempo Libre

que corresponde.

Los resultados del test de las Asignaturas muestran para cada una, la cantidad de estudiantes que seleccionaron cada nivel de agrado. Este test solo permite a los encuestados seleccionar un nivel de agrado para cada asignatura.

Resultados

Cantidad de votos obtenidos por categorías

Asignatura	Me gusta	No me gusta	Meh
Álgebra I	3	1	5
Álgebra II	3	2	1
Análisis Matemático I	1	3	3
Análisis Matemático II	0	0	3
Programación	0	0	3
Filosofía y Sociedad	0	3	0
Lógica	3	0	0
Educación Física I	1	2	6
Educación Física II	12	2	8
Economía Política	12	2	8
Historia de Cuba	10	3	7

Figura 6: Resultados de Asignaturas

---

**2.4 Cómo poner a funcionar la aplicación** La aplicación en su totalidad se compone de dos partes: una en la que se ejecutará el servidor y otra donde se ejecutará el cliente. En nuestro caso, el cliente será la aplicación visual, que consumirá de los recursos del Back-End a través de la comunicación que se establece con la API.

FRONT-END

En la carpeta “App” se encuentra nuestra solución a la aplicación visual y, dentro de esta, se deberá, primeramente, instalar las dependencias.

```
npm install
```

Para “levantar” los servicios de la aplicación visual será necesario ejecutar el siguiente comando:

```
npm start
```

Los resultados se mostrarán en **http://localhost:3000/**.

BACK-END:

***Requisitos:***

- Python: <https://docs.python.org/3/>
- PostgreSQL: <https://www.postgresql.org/docs/12/tutorial-install.html>
- Django Rest Framework: <https://docs.djangoproject.com/en/4.0/topics/install/>

***Ejecución:***

Es necesario, inicialmente, migrar para generar las estructuras de la base de datos correspondiente, para esto deberá ejecutarse:

```
python manage.py makemigrations  
pyhton manage.py migrate
```

Una vez creada la base de datos ya se puede ejecutar la API:

```
python manage.py runserver
```

Si se desea consumir de la API desde un IP diferente se recomienda *ngrok*, para más información consultar [7]. Además se puede utilizar Postman [8] para probar todos los endpoints que contiene la API y fueron descritos anteriormente, para más facilidad el trabajo trae adjunto la colección utilizada por nosotros para el proceso de prueba de la aplicación.



### 3. Anexos

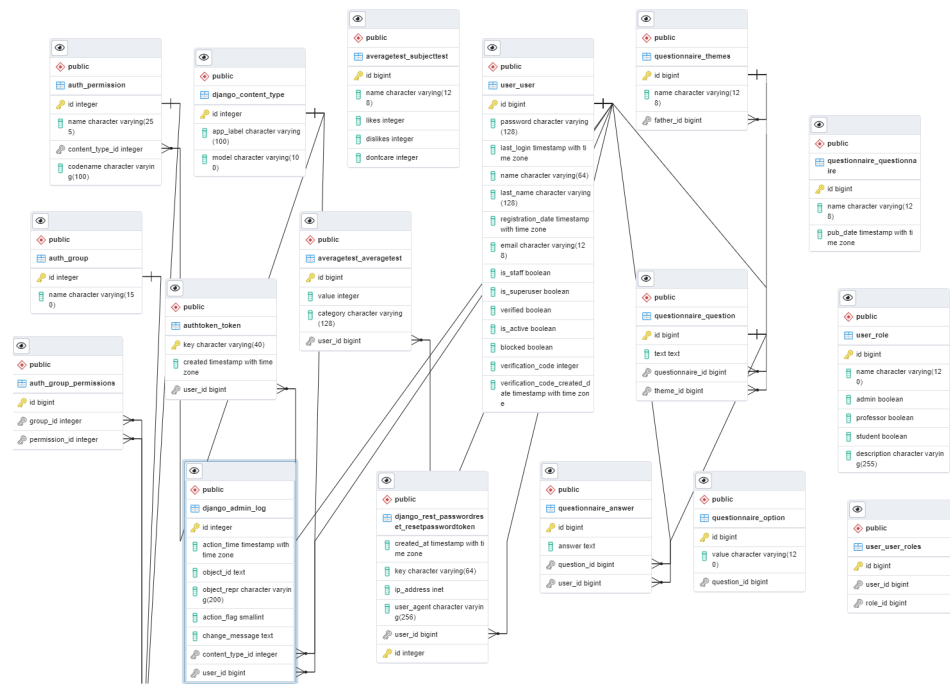


Figura 7: Esquema de la Base de Datos

## ¿Cuán creativo eres?

1. Siempre trabajo con mucha certeza de que estoy siguiendo el procedimiento correcto para resolver un problema en particular. \*

☒ Sí ☐ No lo sé ☐ No

2. Sería una pérdida de tiempo para mí hacer preguntas si no tuviera esperanza de obtener respuestas. \*

☐ Sí ☐ No lo sé ☐ No

3. Me concentro más en lo que me interesa que la mayoría de la gente. \*

☐ Sí ☐ No lo sé ☐ No

4. Siento que un método lógico paso a paso es mejor para resolver problemas. \*

☐ Sí ☐ No lo sé ☐ No

5. En grupos, de vez en cuando expreso opiniones que parecen desanimar a la gente. \*

☐ Sí ☐ No lo sé ☐ No

6. Paso mucho tiempo pensando en lo que los demás piensan de mí. \*

☐ Sí ☐ No lo sé ☐ No

7. Es más importante para mí hacer lo que creo que es correcto que intentar ganarme la aprobación de los demás. \*

☐ Sí ☐ No lo sé ☐ No

8. Las personas que parecen inseguras acerca de las cosas pierden mi respeto. \*

☐ Sí ☐ No lo sé ☐ No

9. Más que otras personas, necesito tener cosas interesantes y emocionantes. \*

Figura 8: Fragmento del Test de Creatividad

## ¿A qué dedicas tu tiempo libre?

Tus Favoritos

6. ¿Cuáles son tus sagas preferidas? \*

<input type="checkbox"/> Todos	<input checked="" type="checkbox"/> Harry Potter	<input type="checkbox"/> El Señor de los Anillos	<input type="checkbox"/> The Hobbit
<input type="checkbox"/> Star Wars	<input type="checkbox"/> The Matrix	<input type="checkbox"/> X-Men	<input type="checkbox"/> The Hunger Games
<input checked="" type="checkbox"/> Universo Marvel	<input checked="" type="checkbox"/> Universo DC	<input type="checkbox"/> Piratas del Caribe	<input checked="" type="checkbox"/> El Padrino
<input type="checkbox"/> Rocky	<input type="checkbox"/> 007	<input type="checkbox"/> Star Trek	<input type="checkbox"/> Indiana Jones
<input type="checkbox"/> Terminator	<input type="checkbox"/> Jurassic Park	<input type="checkbox"/> The Fast & The Furious	<input type="checkbox"/> Saw
<input type="checkbox"/> ¿Otro? ¿Cuál?	<input type="checkbox"/> Ninguno		

7. ¿Cuáles son tus pelis favoritas? \*

<input type="checkbox"/> Todos	<input type="checkbox"/> Forrest Gump	<input type="checkbox"/> Casablanca	<input type="checkbox"/> Intocable
<input type="checkbox"/> The Departed	<input type="checkbox"/> Titanic	<input type="checkbox"/> Gladiator	<input type="checkbox"/> El pianista
<input checked="" type="checkbox"/> León: The Professional	<input type="checkbox"/> Dirty Dancing	<input type="checkbox"/> Pasaite	<input type="checkbox"/> The Green Mile
<input checked="" type="checkbox"/> Saving Private Ryan	<input type="checkbox"/> La vida es bella	<input type="checkbox"/> El silencio de los corderos	<input checked="" type="checkbox"/> Fight Club
<input type="checkbox"/> Pulse Fiction	<input type="checkbox"/> La lista de Schindler	<input type="checkbox"/> Cadena Perpetua	<input type="checkbox"/> Scarface
<input type="checkbox"/> Braveheart	<input type="checkbox"/> Bohemian Rhapsody	<input type="checkbox"/> Django Unchained	<input type="checkbox"/> El resplandor
<input type="checkbox"/> Inception	<input type="checkbox"/> The Imitation Game	<input type="checkbox"/> Inglourious Basterds	<input checked="" type="checkbox"/> ¿Otro? ¿Cuál?
<input type="checkbox"/> Ninguno			Fresa y Chocolate

8. ¿Cuáles mufles adoras? \*

Por favor conteste la pregunta.

<input type="checkbox"/> Todos	<input type="checkbox"/> La Era del Hielo	<input type="checkbox"/> Shrek	<input type="checkbox"/> Toy Story
<input type="checkbox"/> Up	<input type="checkbox"/> Coco	<input type="checkbox"/> Wall-E	<input type="checkbox"/> Brave
<input type="checkbox"/> Frozen	<input type="checkbox"/> Soul	<input type="checkbox"/> El viaje de Chihiro	<input type="checkbox"/> La tumba de las lágrimas

Figura 9: Fragmento del Test de Tiempo Libre

Test del Sentido del Humor
Responda este cuestionario para averiguar cómo califica su "sentido del humor"

1. ¿Ríes a carcajadas? \*

☒ nunca
☐ casi nunca
☐ raramente
☐ a veces

☐ a menudo
☐ casi siempre
☐ siempre

2. ¿Te diviertes mucho? \*

☐ nunca
☐ casi nunca
☐ raramente
☐ a veces

☐ a menudo
☐ casi siempre
☐ siempre

3. ¿Te ríes de tus propios errores? \*

☐ nunca
☐ casi nunca
☐ raramente
☐ a veces

☐ a menudo
☐ casi siempre
☐ siempre

4. ¿Tienes pensamientos divertidos y felices? \*

☐ nunca
☐ casi nunca
☐ raramente
☐ a veces

☐ a menudo
☐ casi siempre
☐ siempre

5. ¿Eres espontáneo? \*

☐ nunca
☐ casi nunca
☐ raramente
☐ a veces

☐ a menudo
☐ casi siempre
☐ siempre

6. ¿Te gusta escuchar y contar chistes? \*

☐ nunca
☐ casi nunca
☐ raramente
☐ a veces

☐ a menudo
☐ casi siempre
☐ siempre

7. ¿Mientras estudias mantienes una actitud relajada? \*

Figura 10: Fragmento del Test de Humor

Sobre las asignaturas cursadas

1. ¿Qué sentimientos te causaron las siguientes asignaturas? \*

Por favor conteste las preguntas en cada fila.

	Me gustó	Me h...	No me gustó
Lógica	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>
Álgebra I	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>
Análisis Matemático I	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>
Programación	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>
Filosofía y Sociedad	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>
Educación Física I	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>
Álgebra II	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>
Análisis Matemático II	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Historia de Cuba	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>
Economía Política	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>
Educación Física II	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>

Completar

Figura 11: Fragmento del Test de Asignaturas

---

## Referencias

1. <https://www.kellogg.northwestern.edu/faculty/uzzi/ftp/page176.html>
2. <https://maconnnc.org/images/LIFE/Humor%20Quiz.pdf>
3. <https://www.imdb.com/chart/top/>
4. [https://es.wikipedia.org/wiki/Anexo:Artistas\\_m%C3%A1s\\_escuchados\\_en\\_Spotify](https://es.wikipedia.org/wiki/Anexo:Artistas_m%C3%A1s_escuchados_en_Spotify)
5. <https://surveyjs.io/>
6. <https://docs.djangoproject.com/en/4.0/>
7. <https://ngrok.com/docs>
8. <https://www.postman.com/>