



UNIVERSIDAD DE CHILE
FACULTAD DE CIENCIAS FÍSICAS Y MATEMÁTICAS
DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN

DISEÑO E IMPLEMENTACIÓN DE APLICACIÓN WEB PARA EVALUACIÓN DE
ALGORITMOS DE RECUPERACIÓN DE OBJETOS 3D

MEMORIA PARA OPTAR AL TÍTULO DE
INGENIERO CIVIL EN COMPUTACIÓN

CLAUDIO ALONSO MALLEA AVILÉS

PROFESOR GUÍA:
IVÁN SIPIRAN MENDOZA

MIEMBROS DE LA COMISIÓN:
NANCY HITSCHFELD KAHLER
PABLO GONZÁLEZ JURE

SANTIAGO DE CHILE
2022

Resumen

Los algoritmos de recuperación por contenido en 3D son métodos que extraen información de archivos multimedia. Para evaluar este tipo de algoritmos es necesario la creación de benchmarks donde se deben definir tres secciones: Un set de datos, un archivo para clasificar estos datos y métricas con los que se evaluarán los algoritmos.

Actualmente, evaluar estos algoritmos se hace de forma lenta, poco organizada y con insuficiente transparencia. Tomando como caso de estudio la Shape Retrieval Challenge [1], entidad encargada de organizar competencias buscando evaluar la efectividad de este tipo de algoritmos, se puede apreciar que organizadores de diferentes competencias están realizando por su cuenta los mismos procesos en la creación de sus benchmarks que otros organizadores, teniendo que definir los sets de datos, los archivos clasificadores y la programación de métricas para sus competencias. Teniendo esto en cuenta, se desarrolló una aplicación web que acelera y facilita estos procesos, donde los participantes pueden obtener sus resultados de forma rápida y automática.

El sistema implementado consiste en una aplicación web que permite la administración y creación de competencias, entregando un espacio para exponerlas a todos los usuarios del sistema que actuarán como competidores. El sistema se encarga de proveer las métricas y funcionalidades para realizar los procesos de evaluación de algoritmos, haciéndose estos procesos de forma automática y quitando la responsabilidad del organizador de realizarlos. Los competidores podrán observar cómo sus algoritmos se comparan con otros de los demás competidores mediante diferentes herramientas.

Se realizó un proceso de validación en el sistema donde se midió la usabilidad y utilidad del sistema. Hubo comunicación con diferentes organizadores que administraron competencias en la Shape Retrieval Challenge, además de otros posibles usuarios de la aplicación, a los que se les mostró las interfaces del sistema entregando tareas que hacer en esta.

Finalmente, los resultados de la validación mediante la realización de entrevistas y encuestas entregan óptimos resultados, en base a esto es posible entregar una herramienta funcional, efectiva y eficaz que utilizarán los organizadores de competencias en recuperación de contenido 3D en el futuro.

Tabla de Contenido

1. Introducción	1
1.1. Objetivos de la memoria	2
1.1.1. Objetivo General	2
1.1.2. Objetivos Específicos	2
1.2. Estructura de la memoria	3
2. Estado del arte	4
2.1. Situación actual	4
2.2. Soluciones actuales posibles	5
2.3. Conceptos técnicos para la evaluación de algoritmos de recuperación de contenido 3D	6
3. Concepción de la solución	10
3.1. Arquitectura del sistema	10
3.1.1. Arquitectura física	10
3.1.2. Arquitectura lógica	11
3.2. Requisitos de la aplicación	12
3.3. Diseño de la base de datos	14
3.4. Formatos de entrega de usuarios	16
3.5. Mockups	17
4. Solución	19
4.1. Herramientas	19

4.2.	Metodología	20
4.3.	Interfaces del sistema	20
4.3.1.	Login	20
4.3.2.	Registro	21
4.3.3.	Contraseña Olvidada	22
4.3.4.	Home	22
4.3.5.	Creación de track	24
4.3.6.	Track	25
4.4.	Estructura de la API	28
4.5.	Inicio de sesión	30
4.6.	Evaluación de entregas de usuarios	31
4.7.	Gráficos del sistema	32
4.8.	Deployment	33
5.	Validación	35
5.1.	Datos utilizados	35
5.2.	Validación de funcionalidad	36
5.3.	Medición de utilidad	37
5.3.1.	Tareas organizadores	37
5.3.2.	Tareas competidores	38
5.3.3.	Resultados del proceso	38
5.4.	Medición de usabilidad	39
5.4.1.	Resultados del proceso	39
5.5.	Comentarios	40
6.	Conclusión	42
6.1.	Lecciones aprendidas	43
6.2.	Posibles trabajos futuros	44

Bibliografía	45
Anexos	47
A. Otras interfaces del sistema	48
B. System Usability Score	58
C. SHREC - 3D Shape Retrieval Challenge	60
D. Estructuras del código	62

Índice de Ilustraciones

2.1. Ejemplo de matriz de distancia.	7
2.2. Gráfico Precision v/s recall de dos algoritmos, el algoritmo “System A” es más eficiente que el otro.	8
3.1. Arquitectura física del sistema	11
3.2. Arquitectura lógica del sistema	11
3.3. Base de datos del sistema.	16
3.4. Mockup de interfaz Home	17
3.5. Mockup de interfaz Creación de competencia	18
3.6. Mockup de interfaz Track	18
4.1. Vista de Login	21
4.2. Vista de Registro	22
4.3. Vista de Home	23
4.4. Vista de Creación de track	24
4.5. Vista de Track	25
4.6. Ejemplo gráfico tipo “Precision-Recall”.	32
4.7. Ejemplo gráfico tipo “Matriz de confusión”.	32
A.1. Vista de registro de nueva cuenta en la aplicación.	48
A.2. Vista para recuperación de contraseña. Se indica ingresar el correo del usuario.	49
A.3. Vista para recuperación de contraseña. Se indica ingresar la nueva contraseña.	49
A.4. Sección Groundtruth en vista: Creación de track.	50

A.5. Sección Problem Description en vista: Creación de track.	50
A.6. Sección Metrics en vista: Creación de track.	51
A.7. Sección Contacts en vista: Creación de track.	51
A.8. Ventana emergente para cambiar configuración de usuario.	52
A.9. Pestaña Leaderboard en vista: Track.	52
A.10.Pestaña Results en vista: Track, accedida por Organizador. Se muestran los botones para exportar gráficos y el gráfico de Precision-Recall de todas las entregas.	53
A.11.Pestaña Results en vista: Track, accedida por Organizador. Listado donde el organizador puede acceder en detalle a los resultados de un competidor. . . .	53
A.12.Pestaña Results en vista: Track, accedida por Competidor.	54
A.13.Pestaña Results en vista: Track, accedida por Competidor.	54
A.14.Pestaña Submissions en vista: Track.	55
A.15.Pestaña Contacts en vista: Track.	55
A.16.Settings de la vista: Track, muestra las cuatro posibles opciones de configuración en la competencia.	56
A.17.Opción Edit Track de Settings.	56
A.18.Opción Leaderboard Settings de Settings.	57
A.19.Opción Result Settings de Settings.	57
A.20.Opción Delete track de Settings.	57
B.1. Formulario enviado a participantes de validación, primeras 5 preguntas. . . .	58
B.2. Formulario enviado a participantes de validación, ultimas 5 preguntas.	59
B.3. Escala de notas al utilizar SUS.	59
C.1. Pantalla inicial actual de SHREC, las competencias se muestran en el listado de la sección inferior de la figura.	60
C.2. Ejemplo de competencia actual en SHREC.	61
D.1. En la figura (a) se muestra la estructura de carpetas del Front-End mostrando los componentes y en la figura (b) se muestra la estructura para el Back-End.	62

Capítulo 1

Introducción

Los algoritmos de recuperación por contenido son un conjunto de métodos que intentan extraer información desde multimedios (audio, imagen, vídeo, texto y datos) para compararlos usando la información propia del medio, sin uso de información externa. Por ejemplo, la obtención de los elementos que se puedan encontrar en una imagen. Dentro de estos algoritmos se encuentran los enfocados en la recuperación de formas en 3D, normalmente representados por archivos que contienen las coordenadas en 3D, posición de las texturas, caras poligonales y otros datos que ayudan a representar el objeto.

Para evaluar estos tipos de algoritmos es necesario el uso de benchmarks. Estos consisten en tres partes: El dataset con la información de los objetos, generalmente dividido en Target y Query, donde Target representa a los objetos que son parte del sistema de base de datos, y Query los objetos que se buscarán en esta. Un ground-truth o clasificación de los datos, que es necesario para conocer que objetos son relevantes en la búsqueda; y además se necesita una metodología de evaluación que está formada por las métricas que se usan para comparar la eficacia de los algoritmos de búsqueda.

El problema que existe en estas evaluaciones es la lentitud, falta de organización e insuficiente transparencia en el proceso. Tomemos como ejemplo la Shape Retrieval Challenge (SHREC), que es el principal caso de estudio del proyecto, esta es una entidad que presenta múltiples competencias con el objetivo de evaluar la efectividad de los algoritmos de recuperación de formas en 3D para diferentes problemas. Cada competencia (o track, que es la forma en la que se refieren a estas) presenta un benchmark con la información correspondiente. El problema de esto es que cada uno de los grupos organizadores de diferentes tracks están haciendo el mismo trabajo, pero de forma individual, es decir, cada grupo entrega un dataset, un ground-truth y algunas métricas de evaluación, pero podemos notar que en la mayoría de los tracks el dataset siempre se divide en Target y Query y las métricas de evaluación son las mismas. Para revisar ejemplos de tracks exitosos a los que posteriormente se hizo un paper con los resultados, se recomienda verificar los siguientes proyectos: *SHREC 2020: Retrieval of digital surfaces with similar geometric reliefs* [2] y *SHREC 2020: Track on multi-domain protein shape retrieval* [3]

Además, cada grupo organizador debe entregar por su cuenta los resultados. Estos tienen que programar las funciones o algoritmos que representen las métricas de evaluación, luego

toman los datos entregados por los competidores, se evalúan y proceden a entregar los resultados a los participantes. Este proceso es muy lento y poco transparente, puesto que los algoritmos usados en las métricas de evaluación son normalmente los mismos entre tracks diferentes. Además de que el competidor no tiene la seguridad de que sus datos están siendo evaluados de manera imparcial, debido a que el grupo organizador tiene toda la responsabilidad de evaluar los resultados por su cuenta. SHREC no ha cambiado la forma en la que se desarrollan estos tracks desde su creación en el año 2007.

Una mejora u optimización en la forma en que se llevan a cabo estos procesos podría garantizar una facilitación en la replicación de experimentos y rapidez en la participación y en la entrega de resultados. Esto ayudaría a la comunidad de recuperación de formas 3D, puesto que, al garantizar una mejor forma de participar en la competencia se estaría acelerando el proceso de mejora en estos algoritmos. De manera que los avances en esta área se harían de manera más rápida y eficiente.

1.1. Objetivos de la memoria

1.1.1. Objetivo General

Este trabajo de memoria busca principalmente acelerar, facilitar y optimizar el proceso de evaluación de algoritmos de recuperación de contenido 3D por medio de la creación de una aplicación web. Se espera que los participantes en la creación de algoritmos reciban los resultados de la evaluación de forma rápida y automática, sin la necesidad de la intervención del grupo organizador a cargo del benchmark. La idea es crear una herramienta que sea útil a la comunidad de SHREC y para todos los grupos interesados que tengan un benchmark de recuperación por contenido de formas 3D.

1.1.2. Objetivos Específicos

Para lograr el objetivo general se plantean los siguientes objetivos específicos, estos corresponden a los procesos necesarios para la creación de la aplicación web.

1. Definir los requisitos funcionales y no funcionales de la aplicación.
2. Generar un modelo de datos que permita representar los benchmarks, los usuarios y sus relaciones.
3. Realizar el diseño de la solución mediante la creación de mockups, ilustrando la idea general de la herramienta.
4. Desarrollar el Back-end de la aplicación, haciendo uso de frameworks, lenguajes y otras herramientas.
5. Desarrollar el Front-end basándose en el diseño anteriormente realizado.
6. Subir la aplicación a un servidor externo.

1.2. Estructura de la memoria

Este informe presenta 6 capítulos. El capítulo 1 “Introducción” indica el problema, la relevancia de este y los objetivos de la memoria. El capítulo 2 “Estado del Arte” consiste en definir todos los conceptos que son necesarios para entender el trabajo realizado, refiriéndose a algoritmos, metodologías, tecnologías y descripción de algunas soluciones. El capítulo 3 “Concepción de la solución” presenta una descripción detallada de la solución teórica del problema, definiendo requisitos y criterios de aceptación. El capítulo 4 “Solución” consiste en describir la solución realizada durante el periodo de memoria, relatando como se realizó la implementación de esta y las características que esta tiene. El capítulo 5 “Validación” presenta como se validó este sistema con los usuarios. Finalmente, el capítulo 6 “Conclusión” describe las conclusiones obtenidas en el desarrollo de este proceso y se describen algunas posibles mejoras al sistema que se podrían realizar en el futuro.

Capítulo 2

Estado del arte

En este capítulo se explicará la situación actual, soluciones actuales posibles y explicación de algunos conceptos que son necesarios para entender cómo funcionan las evaluaciones de algoritmos de recuperación de contenido en 3D.

2.1. Situación actual

Poniéndonos en el contexto de SHREC, el proceso de evaluación transcurre de la siguiente manera: Existe un comité inicial con el que uno se debe comunicar vía correo electrónico para ofrecer una propuesta donde ellos evalúan si es que ésta es aceptada o no en los tracks de SHREC. Una vez que esto se confirma, el equipo organizador debe crear su dataset y su benchmark, para esto se deben crear archivos de clasificación y scripts de evaluación.

La información sobre el track se publica en una página web, donde se anota la descripción del problema y los datos, además de la información de contacto, o sea una dirección de correo. Una vez que se publica la fecha de registro, se publica la data y se explica la metodología indicando que se debe entregar una matriz de distancias. Luego cada equipo participante procede a correr sus experimentos, generando las matrices (que son archivos de texto).

Los competidores pueden enviar más de una matriz de distancias, dando la posibilidad de poder utilizar varios algoritmos o parámetros distintos. Además, se debe indicar una descripción de cada una de las matrices, indicando cómo se obtuvieron. Luego los grupos organizadores agrupan los conjuntos de matrices de los equipos y se procede a generar los resultados utilizando los scripts que se crearon correspondientes a la metodología de evaluación. Una vez que se tienen estos, se les envían por email para cada uno de los competidores.

A veces ocurre que los equipos indican que los resultados no son lo que ellos esperan. Cuando esto ocurre se debe poner en contacto con el grupo organizador correspondiente y verificar si se pueden analizar estos datos. Todo este proceso es lento y depende del tiempo que tengan los grupos organizadores. Finalmente, cuando se tienen los resultados de todos los participantes se procede a escribir un artículo conjunto. Este reporte será entregado a la comunidad de *Workshop on 3D Object Retrieval 2021* [4]. Si el paper es validado y aprobado

por un filtro de revisión estos podrían ser publicados en una revista.

2.2. Soluciones actuales posibles

Con respecto a soluciones actuales que podrían resolver el problema, existen otras herramientas o plataformas que resuelven problemas similares para otros contextos o de forma específica. Este es el caso del *Completion 3D benchmark* [5]. En esta plataforma se busca evaluar métodos para resolver un benchmark con respecto al problema de “*3D Object Point Cloud Completion*”, donde se entrega una serie de puntos incompletos que representan un objeto en 3D y se busca que los métodos entreguen otra serie de puntos representando el objeto en su completitud. Esta plataforma resuelve el problema de la organización del benchmark utilizando un sistema de usuarios donde se puede participar de forma rápida, además de un leaderboard donde se pueden comparar los resultados de las métricas.

Existen otras plataformas que podrían utilizarse para este problema como es el caso de Codalab [6]. Este es un framework para crear o participar en competiciones que requieran ejecutar algún proceso computacional. La mayoría de las competiciones tratan temas de machine learning, pero no se limitan a estas, permitiendo exponer cualquier problema en el que sus soluciones se entreguen por medio de un archivo zip, que será evaluado por métricas que deben devolver un valor numérico, para luego ser expuestos en un leaderboard.

Codalab utiliza un sistema de bundles para crear una competencia. Un bundle es un conjunto de archivos html, py y yaml, que representan todo el contenido de una competencia refiriéndose a: la información del problema, los datasets, la forma en la que se muestran los resultados, entre otras páginas que entregan información. Esto resulta poco intuitivo a la hora de realizar una competencia, debido a que normalmente el organizador descarga un ejemplo de bundle y procede a modificar los archivos que ahí se encuentren para explicar su competencia, por lo que no existe una interfaz que guíe al usuario en la creación de esta.

Esta herramienta resuelve el problema, pero de forma muy general, puesto que no solo se pueden realizar competencias de proyectos de recuperación de objetos 3D (que es el enfoque del proyecto), sino que cualquier tipo de problema que necesite resolver un proceso de forma computacional. Esto impide que la misma herramienta Codalab entregue métricas o scripts que pertenezcan a problemas particulares, debido a la gran variedad de estos. El usuario organizador es el responsable de entregar el código que servirá para computar las entregas de los competidores. Esto es una debilidad que este proyecto planea abordar, puesto que la misma aplicación le permitirá al usuario definir que métricas utilizar, debido a que éstas estarán integradas en la aplicación. El mismo problema ocurre con la muestra de resultados, puesto que también es responsabilidad del organizador definir de qué manera estos serán expuestos.

Otra página web que intenta resolver el problema de forma no enfocada y similar a la herramienta anterior es Kaggle [7]. Esta es una plataforma que permite buscar y subir diferentes tipos de datasets, además de permitir la creación y participación de competencias donde se tenga que resolver un problema de carácter computacional. Dependiendo del tipo de problema, Kaggle ofrece diferentes opciones para la creación de competencias, por lo que el

precio para el hosting del servicio depende del objetivo de estas. Una de estas opciones son las competencias con propósito de investigación; estas no tienen costo. En esta opción se entrega una página con todas las pestañas y opciones que tendrá la competencia como información, leaderboard, reglas, data, entre otros. Cada una de estas pestañas tiene placeholders que informan el contenido que debería llevar cada una de estas. En comparación a Codalab, la forma de crear competencias de Kaggle es más restrictiva, dando una interfaz con opciones limitadas, pero gracias a esto la creación de la competencia resulta más intuitiva. Además, esta plataforma al igual que Codalab no tiene un enfoque con respecto al tipo de problemas a realizar, no entrega métricas o scripts que puedan ayudar a un organizador a definir como se computan las entregas de los competidores o la muestra de los resultados.

2.3. Conceptos técnicos para la evaluación de algoritmos de recuperación de contenido 3D

Este proyecto está enfocado en los algoritmos de recuperación en 3D; estos presentan diferentes maneras para realizar su evaluación de efectividad, es decir, la calidad de la respuesta retornada por estos sistemas. El objetivo de una evaluación de efectividad es medir la habilidad de recuperar objetos relevantes del dataset y de evitar los que no son relevantes. A continuación, se detallarán algunos conceptos para entender estas métricas (formas de evaluar efectividad) y se definirán las cuatro métricas primarias, que son las que se utilizan para la creación de métricas más consolidadas. Además, estas métricas son las que normalmente se usan en competencias de SHREC.

Matriz de distancia

Cuando se tiene un dataset donde se quiere obtener información de este por medio de algoritmos de recuperación, el dataset debe dividirse en Target (Training set) o Query (Test set), donde el Target corresponde al set del dataset con el que el algoritmo será entrenado y el Query el set con el que será evaluado. Cuando uno ejecuta uno de estos algoritmos se obtienen distancias de los objetos del Query a los objetos del Target. De esta forma se puede obtener una matriz de distancia, donde las filas corresponden a todos los objetos del Query, las columnas representan a los objetos del Target y los valores entre una fila y una columna representan la distancia de ese objeto del Query al Target. De esta manera, si uno fuera a ordenar cualquier fila de la matriz, se podrían identificar cuáles son los objetos con menos distancia, es decir, los que el algoritmo determinó que son más similares. En la Figura 2.1 se muestra un ejemplo de matriz de distancia, compuesta por 3 objetos en el Query y 5 objetos en el Target, mostrándose las distancias entre 2 objetos.

Query \ Target	Objeto 1	Objeto 2	Objeto 3	Objeto 4	Objeto 5
Objeto 1	0.12	0.67	0.22	0.23	0.31
Objeto 2	0.78	0.89	0.34	0.80	0.12
Objeto 3	0.13	0.45	0.12	0.99	0.27

Figura 2.1: Ejemplo de matriz de distancia.

Para realizar una evaluación del algoritmo, se toma la matriz de distancia y se evalúa cada una de sus filas, es decir, se evalúa por cada objeto del Query. Luego se realiza un promedio de estos resultados obteniendo valores que definen el rendimiento general del algoritmo.

Objetos relevantes

Para realizar la evaluación se deben identificar los objetos relevantes y no relevantes cuando se analiza una fila de la matriz de distancia. Si un objeto del Query es de la misma clase que la del Target entonces es un objeto relevante.

Es necesario definir que cuando se tiene una lista acotada (una lista donde no se han recuperado o no se han analizado todavía todos los elementos) entonces los elementos se dividen en cuatro grupos: Los elementos “Positivo Correcto”, que son los elementos que se encuentran en esta lista y que son deseados, los “Falsos Positivos” que son los elementos que se encuentran en la lista y que no son deseados, los elementos “Falso negativo”, que son los elementos que no se encontraron en la lista y que eran deseados y los elementos “Falso positivo” que son los elementos no encontrados y que no eran deseados. De esta forma los elementos relevantes corresponden a los “Positivos Correctos” y los no relevantes corresponden a todos los demás.

Precision v/s Recall

Cuando se analiza una fila de la matriz de distancia, primero esta se ordena de forma ascendente, donde los primeros valores tendrán la menor distancia y luego se recorre creando una lista que comienza con solo el elemento que tiene la menor distancia y aumentando la cantidad de elementos de esta lista hasta llegar al máximo (tamaño de objetos en el Target). Luego se pueden definir dos conceptos: Precision y Recall. Precision se define como el cociente de los elementos relevantes entre los elementos recuperados, es decir, la división entre los Positivos Correctos y la cantidad total de elementos analizados hasta el momento. Recall se define como el cociente de elementos relevantes entre el total de elementos relevantes, la división entre los “Positivos Correctos” y la suma de “Positivos correctos” y “Falsos Negativos”.

Luego se puede realizar un gráfico de *Precision v/s Recall*. Estos gráficos muestran como varía la precisión con respecto al recall, es decir, cada punto del gráfico se define como el valor de precisión cuando se recupera cierto porcentaje de elementos relevantes. Si dos algoritmos aparecen en el gráfico de *Precision v/s Recall* y si la curva de un algoritmo está sobre la otra se puede decir que uno es más efectivo que el otro. En la Figura 2.2 se muestra un ejemplo de este tipo de gráficos.

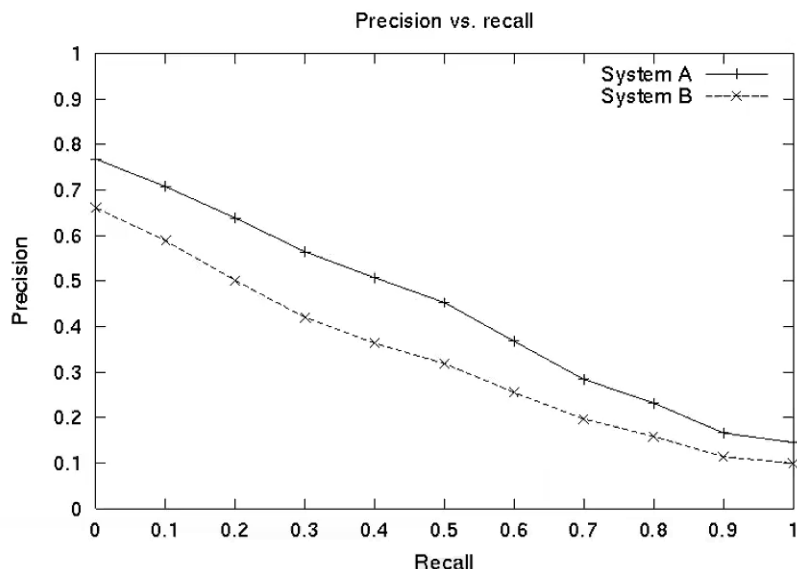


Figura 2.2: Gráfico Precision v/s recall de dos algoritmos, el algoritmo “System A” es más eficiente que el otro.

Métricas

El gráfico de recall-precision presenta un problema, debido a que existen curvas que podrían intersectarse. Esto implica que no se pueda identificar qué algoritmo es más efectivo que otro. Este problema se soluciona con las métricas que se explicarán a continuación, estas buscan representar la información de estos gráficos en un solo número o cantidad. Las métricas son: *Mean Average Precision (MAP)*, *First-Tier (FT)*, *Second-Tier (ST)* y *Nearest Neighbor (NN)*.

La métrica *Mean Average Precision* es una aproximación del promedio de las áreas bajo las curvas de precision-recall. Para una consulta dada, tomando un solo objeto del dataset, por cada elemento relevante se calcula una precisión. Todas estas precisiones se promedian y se obtiene un *Average Precision*. Este proceso se realiza para todos los objetos del dataset obteniendo múltiples *Average Precision* y se procede a realizar un promedio de estos valores, de esta manera se obtiene el *Mean Average Precision*.

First-Tier se define como el valor de la precisión cuando el número de elemento relevantes es igual al número de elementos recuperados, por ejemplo, si se realiza una consulta con 100 elementos en el dataset y 10 objetos relevantes, la precisión se calculará tomando los 10 primeros elementos encontrados en una consulta. La métrica *Second-Tier* es similar, debido a que se define de la misma manera, pero considerando el doble de elementos relevantes.

Finalmente, la métrica *Nearest Neighbor* se define como el valor de la precisión considerando el primer elemento de la lista de elemento recuperados, es decir, el elemento que tuvo la menor distancia de la lista.

Confussion Matrix

La *Confussion Matrix* también es otra técnica para determinar la efectividad de un algoritmo. Estas matrices corresponden a gráficos del tipo heatmap que permiten identificar si el algoritmo está confundiendo una clase con respecto a otra.

Cuando se evalúan todas las filas de la matriz de distancia podemos identificar las clases del Query y Target del valor con menor distancia, el valor con el cual se define la métrica NN. Si el algoritmo es efectivo se espera que las clases sean iguales, es decir, la clase del objeto Query con menor distancia es la misma que la del objeto del Target, pero esto no siempre ocurre por lo que se pueden contabilizar estos valores creando una nueva matriz.

De esta forma se tiene una matriz donde las filas corresponden a las clases del Query y las columnas corresponden a las clases del Target. Entonces el valor de la intersección entre una fila X y una columna Y se puede definir como: “La cantidad de objetos de la clase X del Query donde el valor con menor distancia pertenecía a un objeto del Target de la clase Y”.

Capítulo 3

Concepción de la solución

En este capítulo se describirá la arquitectura del sistema, requisitos de la solución, diseño de la base de datos, formatos que se utilizarán y mockups de la aplicación.

3.1. Arquitectura del sistema

3.1.1. Arquitectura física

En la Figura 3.1 se puede observar la arquitectura física del sistema. Esta se basa en el modelo cliente-servidor, donde el usuario se conecta a través de un navegador mediante la URL de la aplicación al servidor de Amazon Web Services (AWS) [8] donde se encuentra alojada. La interfaz web del usuario se comunica con el Front-End donde se utilizó React [9] para el desarrollo de las interfaces. El Front-End se comunica mediante una REST API con el Back-End, construido con Django [10] y utilizando PostgreSQL [11] para manejar la base de datos. El Front-End también se comunica con un servidor para exportación de gráficos entregado por Highcharts [12] para disponer de la descarga de gráficos en formato PDF o SVG.

Con respecto al servidor de AWS, se utilizó una máquina EC2 [13], el tipo de instancia es t2.micro y el sistema operativo corresponde a Amazon Linux 2 [14]. Las instancias t2.micro corresponden a máquinas que utilizan 1 vCPU y que contienen 1 GiB de RAM. Estas instancias utilizan el servicio Amazon Elastic Block Store (EBS) [15], este es un servicio de almacenamiento diseñado para instancias EC2 y que permiten utilizar 30 GBs de almacenamiento.

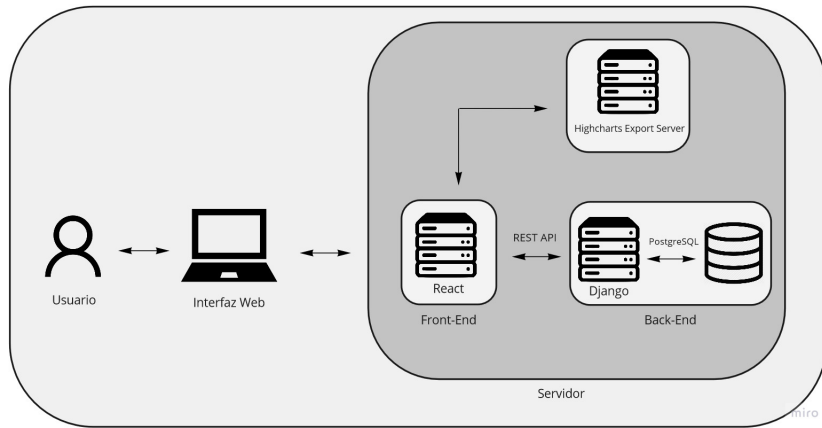


Figura 3.1: Arquitectura física del sistema

3.1.2. Arquitectura lógica

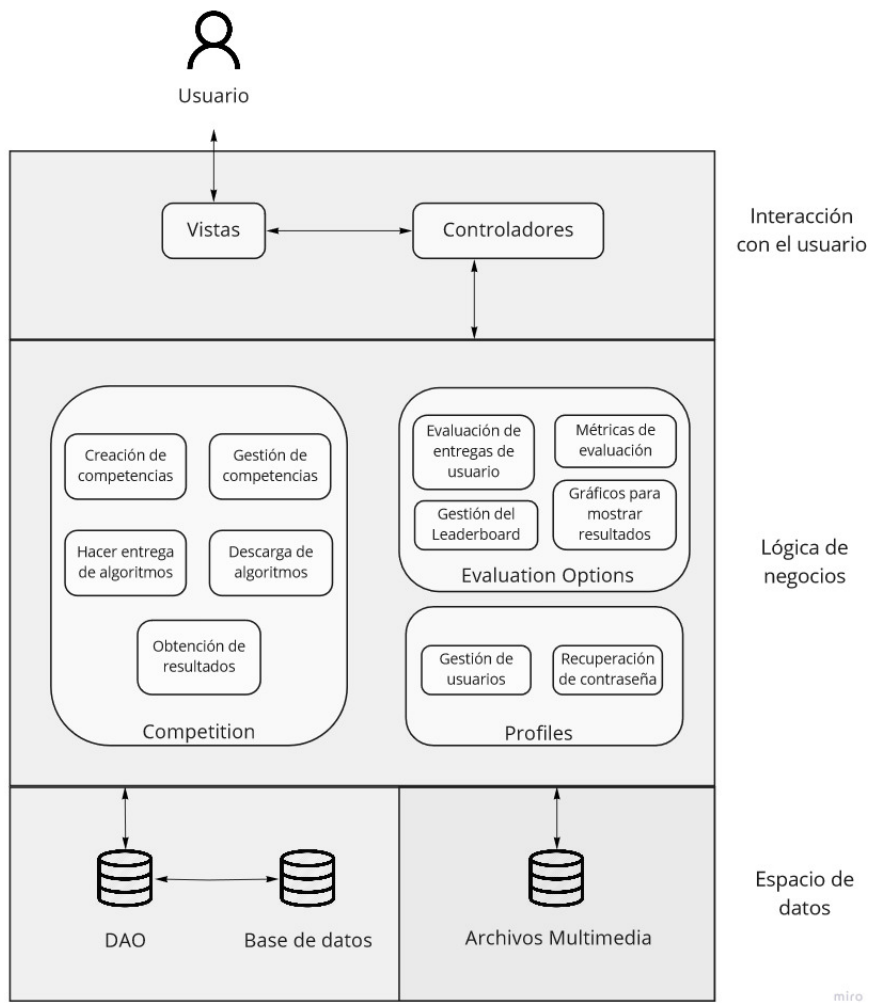


Figura 3.2: Arquitectura lógica del sistema

Como se puede observar en la Figura 3.2 la arquitectura lógica se divide en tres partes: Interacción con el usuario, Lógica de negocios y Espacio de datos. En la Interacción con el usuario una persona interactúa con la aplicación por medio de vistas y estas interactúan con los controladores del sistema.

En la lógica de negocios se observan tres módulos donde se realizan diferentes acciones:

- **Competition:** Este módulo se encarga de realizar todas las acciones referentes a las competencias de la aplicación. Las acciones de este módulo se refieren a la creación y gestión de competencias, la habilidad de los usuarios de hacer entrega y posterior descarga de los algoritmos y la obtención de resultados.
- **Evaluation Options:** En este módulo se manejan la evaluación de algoritmos, gestionando las métricas utilizadas en la evaluación, las configuraciones del Leaderboard y gráficos que mostrarán los resultados.
- **Profiles:** Módulo que maneja la gestión de usuarios, permitiendo la creación de nuevos competidores u organizadores en el sistema y manejando el sistema de recuperación de contraseñas.

Con respecto a la sección Espacio de datos, esta se dividió en dos secciones: La capa de base de datos y la capa de almacenamiento de archivos multimedia. En la primera sección se encuentra la base de datos, la cual corresponde a PostgreSQL, a la que se accede por medio de las funciones y estructuras pertenecientes a Django. En la segunda sección se muestra el almacenamiento de archivos multimedia, estos archivos son guardados de manera local en el servidor donde estará alojada la aplicación.

3.2. Requisitos de la aplicación

Como se mencionó anteriormente, algunos integrantes de grupos organizadores de SHREC proporcionaron ayuda para la realización de este proyecto. Es por esto que se conversó con algunos de estos integrantes para definir cuáles son los problemas que ellos podían identificar en la forma actual en que se realizan las competencias en SHREC. Es por esto que se realizaron una serie de preguntas en un Google Form, para luego ser enviado por correo a los integrantes. Este formulario fue respondido a la brevedad y con los resultados obtenidos se procedió a realizar un informe de requisitos.

Los requisitos son los siguientes:

Requisitos de registro:

- El sistema debe soportar dos tipos de usuarios: Competidores y Organizadores.
- Debe existir un sistema que permita registrarse.
- Debe existir un sistema que permita hacer login.

- Debe existir un botón en el sistema que permita desconectarse de la aplicación en cualquier momento.
- Debe existir alguna ventana donde se pueda ver y modificar la información del usuario.
- El competidores y organizadores deben ingresar estos datos para registrarse: Nombre, Apellido, Correo electrónico y Contraseña
- Los competidores tendrán un seudónimo aleatorio.

Requisitos sobre información de track:

- El sistema debe mostrar la información general del track.
- El sistema debe mostrar cómo está creado y en qué consiste el dataset.
- El sistema debe explicar el problema o challenge del track.
- Se debe explicar el formato de las submissions.
- Se debe permitir ingresar otros tipos de información.
- Se debe dar la opción al grupo organizador de solo entregar un enlace hacia la página del track en SHREC.

Requisitos sobre organizadores:

- Los organizadores deben poder subir un benchmark, lo que implica poder subir: Dataset con la información de los datos, un ground-truth (clasificación de los datos) y métricas de evaluación a utilizar. (Las métricas posibles son: NN, Tiers, nDCG, e-measure, confusion matrices, tier images, precision recall, ROC curves, mAP, entre otros)
- Los scripts con las métricas de evaluación deben ser públicos, estando en un repositorio público de GitHub.
- Debe existir un link en la aplicación que dirija al repositorio de GitHub.
- Los organizadores podrán elegir qué formas de visualizar los resultados (gráficos) necesitan. Gráficos precisión-recall, matrices de confusión, curvas roc, ejemplos de retrieval (Modelo en 3D del resultado).
- Los organizadores deben ser capaces de descargar un archivo con los resultados de los competidores: pdf, png, latex, hojas de cálculo. Dar dos opciones: descargar todos los resultados o descargar uno solo por tabla/gráfico.

Requisitos sobre competidores:

- Los competidores deben ser capaces de seleccionar un benchmark en el cual participar.
- Los competidores deben poder ver toda la información del benchmark (Dataset, ground-truth y métricas).

- Los competidores deben poder descargar los archivos entregados por los grupos organizadores.
- Los competidores deben poder subir sus submissions (matrices de distancias).
- Los competidores deben ser capaces de revisar sus resultados (en formato de gráficos).
- Debe existir un leaderboard con los resultados.
- El leaderboard debe ser anónimo (se mostrará el seudónimo) hasta que se cumpla el deadline de las submissions, al cumplirse la deadline se mostrará el nombre.
- Los competidores tendrán la opción de hacer consultas y dudas, podrán escribir un correo directamente al grupo organizador.

Requisitos sobre seguridad del sistema:

- Debe existir un control de identidad para los usuarios (Tokens).
- El sistema no debe permitir inyecciones de SQL.
- Se debe validar todo input que utilicen los usuarios.
- Debe existir un control sobre las excepciones y errores del back-end

3.3. Diseño de la base de datos

Para crear una aplicación web es necesario tener una base de datos que represente de forma efectiva al sistema. De esta manera, la base de datos debe ser capaz de almacenar distintos tipos de información: Los datos de usuarios, información general de competencias, resultados, métricas y gráficos del sistema. La Figura 3.3 muestra la base de datos creada para la aplicación. A continuación, se mostrará en detalle la estructura que tiene esta base de datos.

Para modelar los tipos de usuario se decidió por tener una tabla llamada Usuario que contenga todos los datos que representan a un usuario en el sistema. Los atributos de la entidad son: Correo, Nombre, Apellido y Contraseña. Para representar que existen diferentes tipos de usuarios en la aplicación, refiriéndose a los competidores y organizadores, se crearon dos tablas con sus respectivos nombres, que referencian a la tabla Usuario. Con respecto a los atributos adicionales que tienen estas tablas, la entidad Competidor tiene el atributo Seudónimo que representa el nombre aleatorio que se le dará al usuario y la entidad Organizador tiene una institución asociada.

La entidad Organizador está relacionada a la entidad Grupo Organizador, este representa el conjunto de organizadores que coordinan la competencia, solo tiene el nombre del grupo como atributo. La entidad Grupo Organizador y la entidad competidor están asociados a la tabla Track, esta tabla representa a las competencias. Utilizando esta relación entre las entidades es posible identificar quienes son los competidores de una competencia en particular y cuál fue el grupo organizador que la creó. Los atributos de la tabla son todos los datos que

tiene un track refiriéndose a: su nombre, la información general de la competencia que sirve de descripción, un link opcional que dirige a la página del track, una imagen representativa, el estado y una llave foránea al organizador creador de la competencia.

La entidad Track se encuentra relacionada a la tabla Benchmark, que representa a todo el proceso e información necesaria para realizar una evaluación de efectividad de los algoritmos. Esta tabla tiene atributos como: texto en Markdown que se utiliza en la página del track, una fecha de inicio y de fin de la competencia, archivos del GroundTruth que representan al dataset como Target y Query, una descripción del formato de entrega y una llave foránea al track correspondiente. En una versión preliminar del modelo se decidió incorporar el atributo Dataset en esta tabla, pero se decidió eliminarla, debido a que el sistema no sería capaz de almacenar todos los dataset debido a su peso, por lo que se prefirió que los organizadores entreguen links en el editor de Markdown para la descarga de estos.

Adicionalmente existen otras entidades llamadas Métricas y Gráficos, la primera representa a las métricas con las que la aplicación evaluará automáticamente los algoritmos de los competidores, sus atributos se componen por: el nombre de la métrica, un nombre más corto que sera utilizado en el leaderboard y un link a GitHub donde se encuentra el código de manera pública (esto se hace debido a uno de los requisitos que estipula que las métricas deben ser transparentes). La segunda entidad mencionada Gráficos representa a todos los gráficos y formas en las que se mostrarán los resultados de los participantes. Estas dos entidades están relacionadas a otras tablas, las cuales son: Selección_Métricas y Selección_Gráficos, estas tablas sirven para representar la selección de todas las métricas y gráficos.

También existe la tabla LeaderBoard.Settings. Dentro de cada competencia existe un leaderboard que muestra los resultados de los competidores, ordenándolos por efectividad de forma descendente. Los organizadores deben definir por cual métrica se ordenarán estos resultados y además pueden definir que métricas aparecerán efectivamente en este ranking. Es por esto que la tabla tiene el atributo de “order_by_metric” que representa la métrica con la que se ordenará, también tiene una llave foránea a la tabla Benchmark y además la tabla tiene una relación 1 a N con la tabla Métricas puesto a que corresponde a la selección de métricas que aparecerán en el Leaderboard.

Los organizadores tienen la capacidad de editar en su competencia las entregas de los usuarios. Es por esto que existe la tabla Acronym, que tiene de atributo “acronym” para representar el nombre que le quiera dar el organizador a la entrega del usuario. Este es un acrónimo debido a la tendencia de los organizadores a cambiar los nombres de las entregas de los usuarios por una sigla de un tamaño reducido. Esta tabla se relaciona con las tablas Track y User, para determinar la competencia y usuarios correspondientes, por lo que tiene llaves foráneas de estas tablas.

Un competidor puede hacer entrega de sus algoritmos en el formato pedido por el organizador. Es por esto que existe la tabla Submissions, que tiene de atributos: el archivo entregado por el usuario, el número de intento que se utilizó para subirlo y dos llaves foráneas, una a la tabla Competidor y la otra a Track.

Finalmente, se tiene la tabla Resultados, esta representa a los resultados de los competidores luego de que el sistema haya hecho evaluación de sus entregas con respecto a un track

en particular, es por esta razón que la tabla se relaciona con Competidor y Track. El atributo de esta entidad es un JSON donde se almacenarán los resultados.

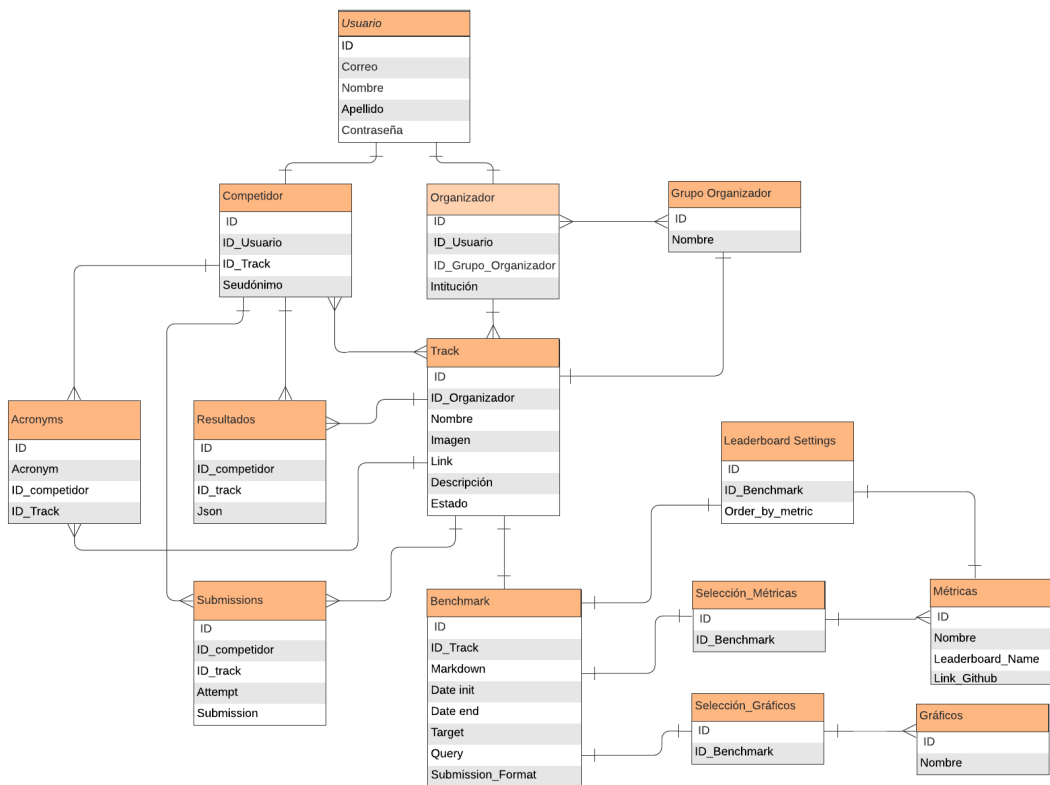


Figura 3.3: Base de datos del sistema.

3.4. Formatos de entrega de usuarios

En las competencias de *SHREC* los organizadores tienen la libertad de pedir a los usuarios un tipo de formato específico para la entrega de algoritmos. Estos pueden ser el código de los algoritmos junto a la instrucción de como ejecutarse, matrices de distancia que representan lo que se obtiene al ejecutar el algoritmo o hasta archivos *.txt* representando los resultados del algoritmo en el formato que el organizador indique.

Como esta aplicación web busca automatizar estos procesos para facilitar el trabajo que debe realizar el organizador, se decidió por utilizar matrices de distancia dado que es un formato estándar ocupado por la comunidad para evaluar los algoritmos. Si se decidiera que los usuarios entreguen el código de sus algoritmos no existiría una forma para automatizar estos procesos en el sistema.

Estas matrices de distancia corresponden a archivos *.txt*, donde existe una cantidad de líneas correspondientes al número de objetos en el *Query* donde cada línea contiene un número de valores igual al de objetos en el *Target* separados por espacios. De esta manera, una línea en el archivo representa el conjunto de distancias obtenidos por el algoritmo con respecto de un objeto del *Query* en comparación a todos los objetos del *Target*. El orden

en el que se escriben estas filas y columnas corresponden al orden que indiquen los nombres de los objetos en los dataset correspondientes. Para ejemplificar, si el dataset *Target* tiene un archivo llamado 3.obj y el dataset *Query* tiene un archivo llamado 5.obj entonces en la matriz de distancia, el valor en la quinta fila en la tercera columna se encontrará la distancia del objeto del archivo 5.obj del *Query* en comparación al objeto del archivo 3.obj del *Target*.

3.5. Mockups

Antes de la implementación de la solución se crearon mockups o diseños de baja fidelidad que representan la forma general que tendría la aplicación luego de su implementación sin tener en cuenta detalles como la estética de esta, sino que centrándose en qué tipo de información aparecerá en la aplicación y el espacio o posición donde se encontrará. Para esto se diseñaron múltiples vistas utilizando la herramienta Balsamiq [16]. A continuación, se mostrarán algunos de los diseños más importantes de esta etapa del proceso.

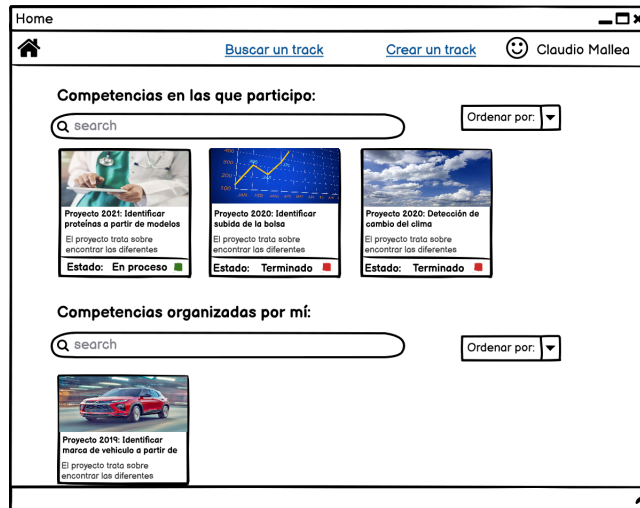


Figura 3.4: Mockup de interfaz Home

En la Figura 3.4 se puede observar la página de inicio que se planteó para el sistema. El principal objetivo de esta página Home es mostrar a los usuarios de la plataforma cuáles son sus competencias, diferenciando las competencias en las que participan y las que están organizando. El elemento que representará a la competencia debe ser identificable, se deben observar diferentes datos característicos de las competencias, como su título e imagen, pero sin mostrar demasiada información que agobie al usuario. Es por esto que las competencias se muestran en un formato carta, un elemento pequeño que puede mostrar esta información.

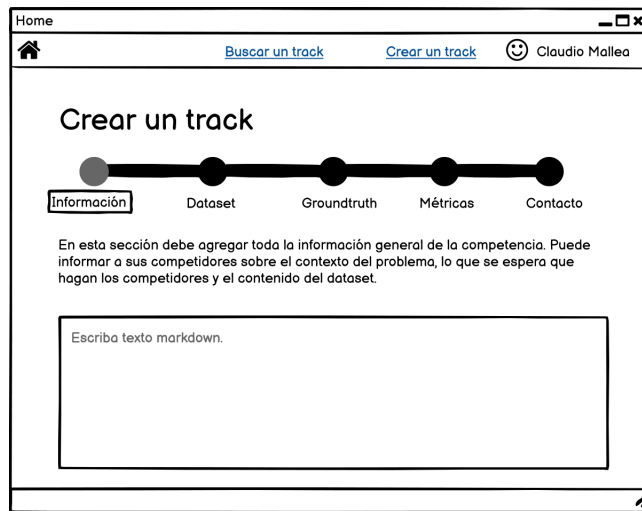


Figura 3.5: Mockup de interfaz Creación de competencia

En la Figura 3.5 se muestra la interfaz que servirá para que los usuarios creen sus propias competencias. Se les mostrará a los usuarios un formulario separado en diferentes secciones para que puedan incluir los datos necesarios para la competencia.



Figura 3.6: Mockup de interfaz Track

En la Figura 3.6 se presenta la página que contiene a la competencia. Esta es la página donde los competidores podrán ver la información completa del track, teniendo la opción de participar haciendo entrega de sus algoritmos, además de poder ver sus resultados junto a los organizadores. Se pensó en separar la página en dos secciones, en la parte superior se mostrará un banner que sirve como extensión del elemento carta mostrado en la página de inicio, pero con toda la información visible. También se muestra una sección inferior donde se encuentran separadas en pestañas todas las funcionalidades de la competencia.

Capítulo 4

Solución

En este capítulo se explicará la implementación de la solución, revisando las interfaces del sistema, las peticiones implementadas, implementación de funciones importantes y el deployment de la aplicación.

4.1. Herramientas

Como la aplicación se encuentra segmentada en un Back-End y un Front-End se hizo uso de diferentes herramientas para la creación de estos. En el Back-end se utilizó Django para la creación de modelos, vistas y scripts escritos en Python, adicionalmente se utilizó Django REST framework [17] para realizar la API del sistema. Se utilizó Django para crear la aplicación debido a que el estudiante está familiarizado con este framework, habiéndolo utilizado en diferentes ocasiones en la carrera.

Para el front-end se decidió utilizar React.js. Esta es una librería open source de JavaScript para la creación de interfaces de usuario que divide las interfaces en componentes que pueden ser reutilizados de forma intuitiva. Esta librería fue utilizada por el memorista debido a que ya había tenido experiencia con esta. Además, comparado con otras herramientas como Vue.js o Angular que son frameworks, esta herramienta al ser una librería resulta más intuitiva y rápida de aprender, debido a que tiene un rango de funcionalidades de menor tamaño. Esto es relevante debido a que esta aplicación debe realizarse en un tiempo acotado, por lo que disminuir el tiempo de aprendizaje de la herramienta resulta valioso. Junto a React.js se utilizó React-Bootstrap [18], una librería que permite utilizar las clases de Bootstrap como componentes en React.

En la fase de deployment se utilizó Docker [19] para subir el proyecto a un servidor. Docker es un framework que permite separar la aplicación de la infraestructura en la que está instalada mediante el uso de contenedores, funcionando sin problemas en cualquier máquina independiente del sistema operativo. La utilización de este framework fue opcional debido a que es posible subir la aplicación sin el uso de contenedores en un servidor externo, pero el uso de este facilita la instalación de librerías que aceleró el proceso de deployment de manera

considerable.

4.2. Metodología

Junto al profesor guía se decidió que para abordar la realización del sistema este se dividiría en varias funcionalidades y durante los viernes de cada semana se tendría una reunión para evaluar el avance de estas. De esta manera se pudo trabajar en el Front-end y el Back-end al mismo tiempo, avanzando a la siguiente funcionalidad cuando la anterior estuviera terminada.

Se utilizó el software GitHub [20] para subir y tener las versiones de este sistema durante todo el proceso. Se crearon 3 repositorios, uno que tuviera el Back-end creado en Django, otro que tuviera el Front-end realizado con React y otro repositorio que tuviera la versión del sistema que posteriormente se subiría al servidor de AWS, este repositorio fue necesario debido a que existen parámetros y variables de Docker que no son necesarios en los repositorios anteriores.

Después de completar todas las funcionalidades del sistema, las que serán detalladas a continuación, se procedió a deployar este sistema en Docker a una máquina de AWS.

4.3. Interfaces del sistema

La aplicación web se dividió en varias secciones correspondientes a diferentes vistas en el sistema. A continuación, se procederá a detallar los componentes de cada una de estas interfaces, definiendo sus funcionalidades, posición y espacio donde se encuentran.

4.3.1. Login

Esta sección del sistema corresponde a todas las vistas utilizadas para que el usuario pueda acceder o registrarse en el sistema. El Login es lo primero que ve el usuario al acceder por primera vez a la página por lo que se decidió que el diseño fuera simplista, no mostrando tantas funcionalidades en pantalla.

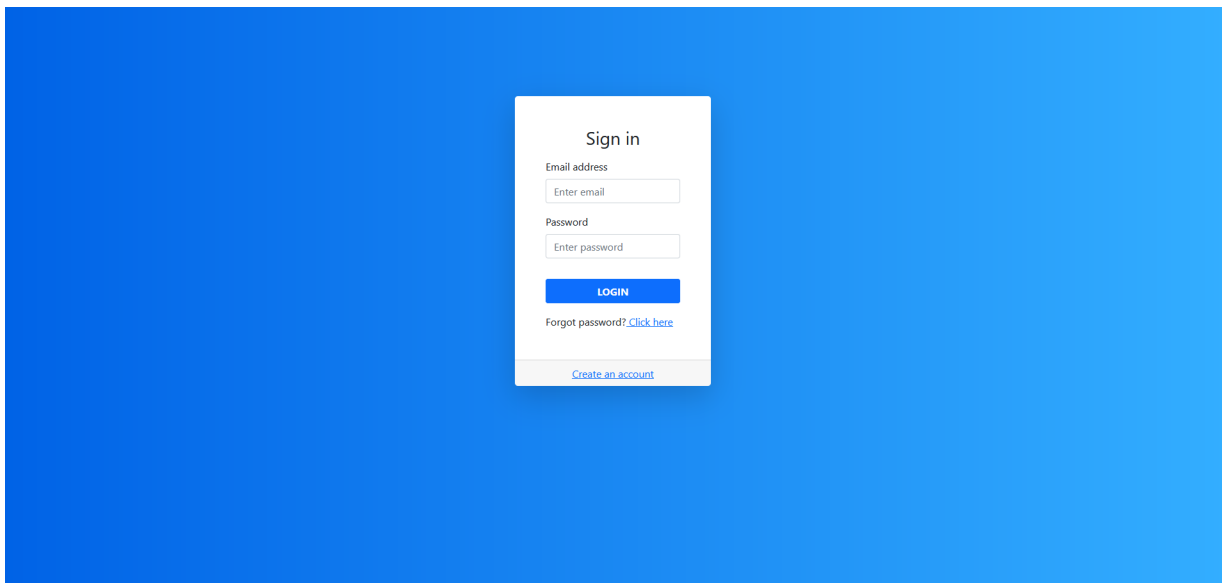


Figura 4.1: Vista de Login

Como se puede ver en la Figura 4.1, esta sección presenta un bloque en el centro de la pantalla con las funcionalidades de la interfaz. Se muestran dos elementos inputs, una sección “Email Adress” donde el usuario coloca su correo que actúa como nombre de usuario en el sistema y una sección “Password” que pide la contraseña del usuario, esta se muestra censurada al escribir. Debajo de estos inputs se presenta un botón Login, si el usuario escribió sus credenciales correctamente podrá acceder al sistema y al contrario se muestra un error indicándolo.

En la parte inferior del bloque se muestra un link para que el usuario pueda recuperar su contraseña si es que a este se le ha olvidado, lo que ve el usuario al hacer clic sobre este link se detallará posteriormente. Además, en la sección inferior que se encuentra al final, hay un link para acceder a la sección de registro.

4.3.2. Registro

En la Figura 4.2 se muestra la vista de registro de la aplicación. En esta vista el usuario puede crear una nueva cuenta en el sistema ingresando sus credenciales, estas son: nombre, apellido, email y contraseña. Las contraseñas deben tener una cantidad mínima de 8 caracteres, deben tener letras y números y no deben contener datos del nombre, apellido o correos, se muestra un “tooltip” donde el usuario puede ver esta información. Para la implementación de este sistema de registro se utilizaron las funcionalidades preconstruidas que entrega Django para crear estos servicios.

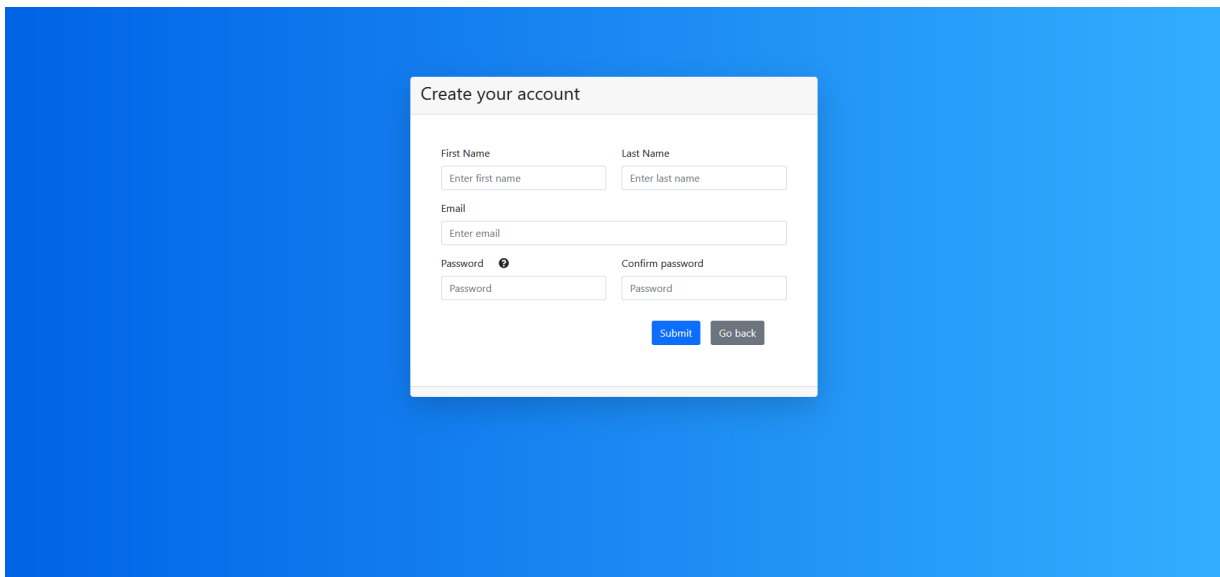


Figura 4.2: Vista de Registro

4.3.3. Contraseña Olvidada

Esta sección del sistema se refiere a todas las vistas que son utilizadas para la recuperación de una contraseña si es que esta ha sido olvidada por el usuario. Se puede acceder a esta sección usando el enlace “Forgot your password?” del login. Al usuario se le pide introducir su dirección de correo, a este se le enviará un mensaje con un link con el que pueden acceder a una nueva vista del sistema. En esta vista se les pedirá introducir una nueva contraseña con los mismos requisitos que se piden en el registro. Si el usuario sigue los pasos correctamente podrá obtener una nueva contraseña con la que podrá entrar al sistema.

Por seguridad, el link enviado al correo del usuario solo puede ser utilizado una vez, esto se debe a que el sistema de reinicio de contraseña funciona mediante tokens que se invalidan una vez que el usuario cambia de contraseña. Para el envío de correos se utiliza un SMTP de Gmail utilizando una cuenta de correo del memorista.

4.3.4. Home

Cuando los usuarios ingresan sus credenciales y logran ingresar al sistema verán la página Home, que se puede observar en la Figura 4.3. Esta vista sirve como nexos para dirigirse a las otras funcionalidades de la aplicación.

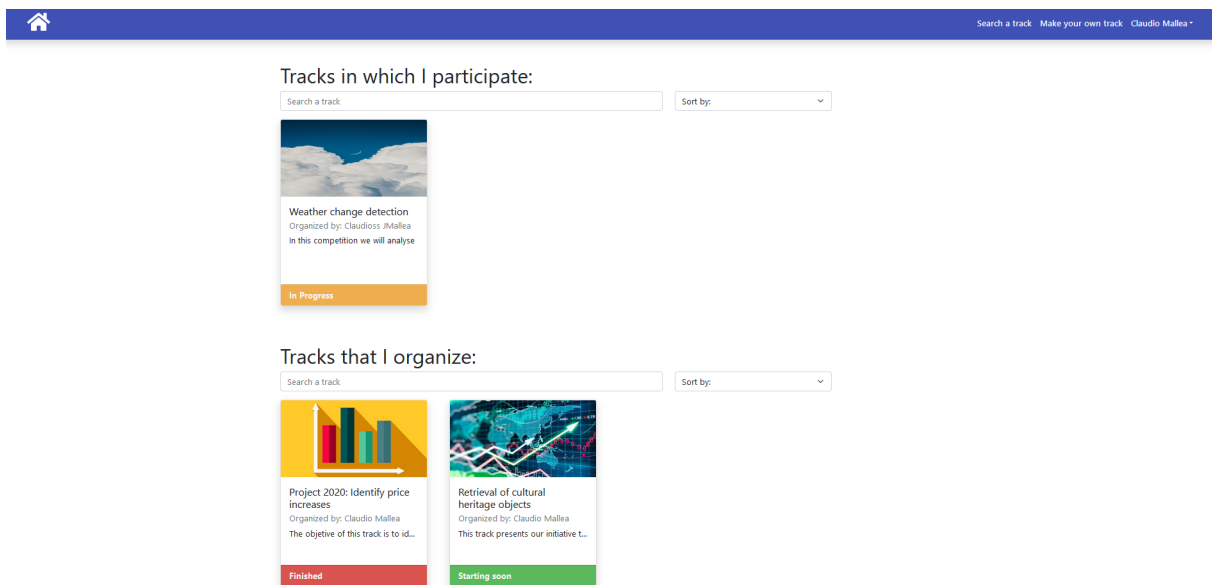


Figura 4.3: Vista de Home

En el centro de esta página el usuario puede ver dos secciones, una sección para competencias en las que está participando actualmente y una sección para competencias en las que él es el organizador. Cada una de estas secciones tiene un buscador y un input para ordenar las competencias del usuario, estas se pueden ordenar alfabéticamente y por fecha. Las competencias aparecen en un formato de carta donde se puede ver la imagen que representa a esta, su título, el inicio de una descripción, el organizador de la competencia y el estado en el que esta se encuentra representada por un color. Estos contenedores son seleccionables por el usuario para poder dirigirse a otra vista donde se encuentra la competencia.

Para obtener estas competencias se realizan peticiones al servidor obteniendo las competencias de la tabla Track según las variables que el usuario escoja. Las peticiones retornan las competencias luego de haber hecho un filtro que determina si el usuario participa o organiza la competencia, los filtros que se hacen cuando se escribe algo en el buscador se hacen directamente en el navegador, por el Front-end.

En la parte superior de esta vista se puede observar la barra de navegación del sistema. Esta barra está disponible para todas las vistas que el usuario accede cuando ya ha hecho inicio de sesión. Esta barra de navegación tiene el objetivo de dirigir a los usuarios a diferentes secciones de la aplicación.

Se muestran tres vínculos en la parte derecha: “Search a track” que dirige a una página similar a Home, pero que contiene un solo buscador que muestra todas las competencias del sistema, “Make your own track” que es una vista donde podrán crear su propia competencia y por último se muestra el nombre de usuario que si es seleccionado aparecerán links para cambiar la configuración del usuario y otro link para desconectarse.

Al seleccionar el link de configuración se muestra una ventana emergente donde aparecen inputs para cambiar los siguientes atributos del usuario: Nombre, Apellido e Institución. En la parte inferior de la ventana se encuentra un botón para hacer subida de estos cambios.

4.3.5. Creación de track

Cuando el usuario selecciona “Make your own track” en la barra de navegación, el usuario se dirigirá a la vista para la creación de su propia competencia. Esta se puede observar en la Figura 4.4. En esta sección se le presentará al usuario distintas secciones donde podrá introducir los datos necesarios para tener su propia competencia en el sistema. La vista se compone de un recuadro al centro de la pantalla donde estarán los inputs para introducir la información y en la parte superior se puede observar una barra de progreso que le informa al usuario en que parte del proceso se encuentra.

The screenshot shows a web interface for creating a track. At the top, there is a dark blue navigation bar with a home icon on the left and the text "Search a track Make your own track Claudio Mallea" on the right. Below the navigation bar is a progress indicator with five steps: "General Information" (active), "Groundtruth", "Problem Description", "Metrics", and "Contact". The main content area is a white box titled "General Information" with a light blue border. Inside the box, there is a paragraph of instructions: "In this section you can write the information that will represent the track on the website. If you already have a website where you explain the problem/challenge to your competitors you can put the link here. The start date represents the day when the track begins, you can't edit important track information after this period, the end date represents the day when the track stops and when your competitors can no longer send their submissions." Below this are several form fields: "Title of track" with a text input and a note "This field must be 8-120 characters long."; "Link (Optional)" with a text input and a note "If you have already a page with the problem information, put it here."; "Short description" with a text input and a note "This field must be 8-250 characters long."; "Start Date:" and "End Date:" each with a "Click to select date" button; and "Picture:" with a file upload area showing "Examinar..." and "Ningún archivo seleccionado." and a note "This picture will identify your track among others." At the bottom right of the form box are "Back" and "Next" buttons.

Figura 4.4: Vista de Creación de track

Esta vista se divide en 5 secciones: Información general, Groundtruth, Descripción del problema, Métricas y Contacto. En la sección de Información General el usuario podrá ingresar los datos que representarán a la competencia en el sistema. Al usuario se le piden 6 datos: el nombre del track, descripción del track, un link opcional al sitio web si es que ya tiene uno donde se explique la competencia, fecha de inicio y de término de la competencia y una imagen opcional. Si el usuario no ingresa una imagen para representar a la competencia se le asigna una por defecto.

En la siguiente sección Groundtruth, al usuario se le pide ingresar los archivos que representan al dataset. Los inputs de este formulario son los archivos Target y Query que deben estar en el formato “The Princeton Shape Benchmark Classification Format” [21] indicado en el enunciado. Adicionalmente en la sección inferior se encuentra un input deshabilitado, este input muestra el formato en que los usuarios deben subir sus entregas, pero como el único formato permitido en el sistema es de matrices de distancia, entonces se encuentra deshabilitado dando a entender que podría cambiar en el futuro.

En la siguiente sección Descripción del problema se les presenta a los usuarios un editor de Markdown y se les informa que pueden ingresar toda la información que ellos estimen necesaria. Esta información debería ser una descripción del problema en detalle, una descripción

del dataset que se va a utilizar y los links de este. A los usuarios se les da la libertad total de escribir lo que ellos estimen conveniente, por lo que también pueden dejar este espacio en blanco y dirigirlos al link donde tienen la información. Este texto en Markdown será lo primero que se vea cuando los competidores entren a la competencia.

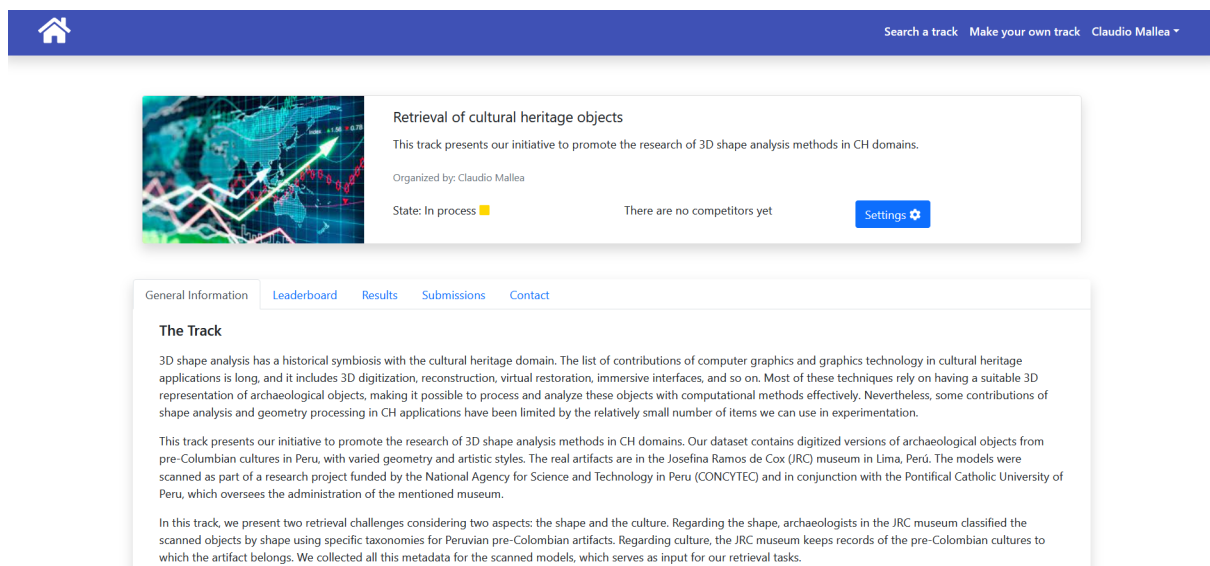
En la siguiente sección Métricas a los organizadores se les presenta un listado de posibles métricas a elegir que serán utilizadas para evaluar las entregas de los usuarios. Estas métricas son: *First Tier*, *Second Tier*, *Nearest Neighbor*, *Mean Average Precision* y *Normalized Discounted Cumulative Gain*. Junto a cada métrica se encuentra un link al repositorio de GitHub donde se encuentra el código de cada algoritmo.

Finalmente, en la última sección Contacto a los usuarios se les presenta un input donde podrán colocar los correos de los usuarios que forman parte del grupo organizador. No existe un límite de correos y los usuarios que estén registrados en el sistema tendrán permisos de organizador en el sistema.

Al finalizar la sección de contacto, se le preguntará al usuario si está seguro de esta configuración, si el usuario acepta, la competencia se creará. Al finalizar, se le presentan dos links al usuario, uno para dirigirse a la competencia o volver al Home.

4.3.6. Track

En la Figura 4.5 se muestra la vista a la que se dirigen los usuarios al seleccionar una competencia en Home. En esta se presenta toda la información del track que los organizadores entregaron en la creación de esta. La vista se divide en dos secciones, una parte superior que presenta un banner y una parte inferior donde los competidores pueden interactuar con las diferentes partes de la competencia.



The screenshot shows a web interface for a competition track. At the top is a dark blue navigation bar with a home icon on the left and search, track creation, and user profile options on the right. Below this is a white card for the track 'Retrieval of cultural heritage objects'. The card features a banner image of a globe with data lines, a title, a description, the organizer's name (Claudio Mallea), a state indicator (In process), a note that there are no competitors yet, and a settings button. Below the card is a detailed description of the track, including its purpose, the dataset source (Josefina Ramos de Cox museum), and the challenges presented.

Figura 4.5: Vista de Track

En el banner se les presenta a los usuarios la imagen de la competencia, el título, la

descripción, el organizador, el estado, el número de competidores y un botón para participar. Los estados pueden ser 3 dependiendo de las fechas de inicio y de fin: Starting si es que la fecha actual es anterior a la fecha de inicio, In progress si la fecha actual está entre la fecha de inicio y de fin, y Finished si es que la fecha es posterior a la fecha de fin, los permisos y acciones disponibles dependiendo de estos 3 estados serán explicados posteriormente.

En la sección inferior de la competencia se puede encontrar una serie de pestañas donde el competidor puede dirigirse para ver las partes de la competencia, estas son: General Information, Leaderboard, Results, Submissions y Contact.

General Information

En General information los usuarios podrán ver el texto en Markdown que el organizador puso a disposición. Esta pestaña está abierta por defecto al entrar en la competencia, por lo que será lo primero que vean los competidores.

Leaderboard

En la sección de Leaderboard se les muestra a los competidores y organizadores una lista con los rankings de los usuarios, mostrando sus nombres y los resultados según las métricas elegidas por el organizador. En la sección superior a la tabla se les informa sobre la métrica con la cual el leaderboard está ordenado. También existen dos botones donde se le permite al usuario descargar la tabla en formato CSV o Látex.

Algunas secciones del track varían según el usuario y el estado en el que la competencia se encuentre, el leaderboard es una de estas secciones. Si el estado del track es Starting se muestra un texto indicando que la competencia aun no comienza por lo que no hay resultados que mostrar. Si el estado es In Progress o Finished a los organizadores y competidores se les muestra la misma información, que es el leaderboard mostrando de forma ordenada los resultados, la diferencia es que los competidores verán seudónimos en vez de los nombres de los usuarios hasta que el estado sea Finished, esto se hace para preservar anonimidad en el proceso.

Results

En la siguiente pestaña Results, se encuentran los resultados de los usuarios utilizando los gráficos que seleccionó el organizador. Los tipos de gráficos existentes en el sistema son: Gráfico de Precision-Recall y Matriz de confusión.

Esta vista, al igual que el leaderboard, varía según el estado y el tipo de usuario. En el estado Starting solo se observa un texto informativo diciendo que la competencia no ha comenzado. Cuando el estado es In Progress o Finished, los organizadores podrán observar un texto informando que pueden cambiar algunas configuraciones usando el botón de Settings en el banner, dos botones para descargar todos los gráficos al mismo tiempo y descargar todas

las entregas de los usuarios, un gráfico de Precision-Recall que compara todas las entregas de todos los usuarios, y en la sección inferior se encuentra un listado de todos los usuarios donde podrán verse los resultados por separado, en este se podrán ver las matrices de confusión por entrega y un gráfico de Precision-Recall con todas las entregas del usuario.

Con respecto a los competidores, a estos se les informa cual fue su mejor entrega según la métrica elegida por los organizadores y se les muestra solos sus resultados con los gráficos anteriormente mencionados.

Submissions

En la pestaña de Submissions, los participantes podrán hacer entrega de sus algoritmos con el formato de matriz de distancia. En esta sección se informan: las fechas de inicio y de fin de la competencia, el máximo número de intentos, una descripción del formato de matriz de distancia, las métricas con las que se evaluarán las entregas y una sección de entregas donde los usuarios pueden subir sus archivos y ver un historial de entregas. Los participantes solo pueden hacer entrega de sus archivos si el estado de la competencia es In Progress, en los demás estados la sección de entregas se encuentra deshabilitada.

Contacts

Finalmente, en la pestaña de Contacts, los usuarios pueden revisar cuales son los correos electrónicos del grupo organizador de la competencia, pudiendo observar quien es el administrador y quienes son los demás organizadores. En un principio se pensó en disponer de un sistema para que los participantes pudieran mandar correos a los organizadores, pero se decidió que esto no se iba a realizar por temas de tiempo.

Configuración del Track

Cuando un organizador que es administrador accede a una de sus competencias, en el lugar donde los participantes pueden participar por medio de un botón el administrador se encontrará con un botón de configuración de la competencia. Este botón abre una ventana emergente con 4 posibles opciones: *Edit track*, *Leaderboard*, *Results* y *Delete Track*. Estas opciones se pueden observar en la Figura A.16 del anexo.

En la sección de *Edit track* el administrador se encuentra con un formulario similar al que tuvo que realizar en la creación del track. Se mostrarán todos los datos que se configuraron en la creación y se le da la opción de editarlos. Sin embargo, si el estado de la competencia dejó de ser Starting, entonces el usuario no puede editar las métricas del sistema, puesto que la competencia ya está en curso.

En la sección de *Leaderboard*, el organizador podrá editar opciones de la pestaña Results de la competencia. El organizador podrá decidir cuáles son las métricas que se mostrarán en el leaderboard, por defecto aparecen todas las métricas seleccionadas en la creación, pero puede

decidir si no mostrar algunas, con la condición de haber por lo menos una seleccionada. En esta sección el usuario también podrá editar la métrica con la que se ordena el leaderboard, por defecto es la métrica MAP si es que se encuentra disponible, en el caso contrario, se mostrará la primera de la lista de métricas disponibles.

En la sección de *Results*, los usuarios podrán editar opciones referentes a la pestaña Results de la competencia. En esta sección los organizadores tendrán la opción de editar cuales son los gráficos a los que los competidores tendrán acceso. Además, podrán cambiar los nombres de las entregas de los usuarios, a través de una lista donde se muestran todos los participantes de la competencia. Estos nombres que en el sistema se les llama “acronyms” solo cambian la vista del administrador, esto les sirve para crear gráficos con nombres que a estos les convengan.

Finalmente en la sección *Delete track* al usuario se le pregunta si desea eliminar la competencia del sistema. Al presionar el botón correspondiente se pregunta al usuario si está seguro de esta acción, cuando esto se confirma la competencia es eliminada.

4.4. Estructura de la API

El Back-End dispone de una API con el que el Front-End puede acceder una serie de peticiones. Estas peticiones actúan con la base de datos para disponer de los diferentes recursos que se necesiten. A continuación, se listarán los 26 endpoints que se crearon para la API de la aplicación, definiendo sus funciones y con qué tablas interactúan de la base de datos según corresponda, además de otras características.

Endpoints del Módulo Evaluation Options:

Estos endpoints interactúan con las tablas que tienen información sobre las formas de evaluar una competencia, estas tablas son: Metrics, Graphics y Leaderboard_Settings

- **GET evaluation-options/metrics/** Utilizada en Vista: Creación de track, Obtiene un listado de todas las métricas en la tabla Metrics.
- **GET evaluation-options/graphics/** Utilizada en Vista: Track, Obtiene un listado de todas las métricas en la tabla Graphics.
- **GET evaluation-options/leaderboardSettings/pk/** Utilizada en Vista: Track, Obtiene la configuración del leaderboard de una competencia en particular.
- **POST evaluation-options/leaderboardSettingEdit/** Utilizada en Vista: Track, Edita la configuración del leaderboard de una competencia en particular.
- **PUT evaluation-options/graphicsSelectionEdit/** Utilizada en Vista: Track, Edita la selección de gráficos utilizados en una competencia en particular

Endpoints del Módulo Profiles:

Estos endpoints interactúan con las tablas que presentan datos sobre los usuarios: User, Competitor, Organizer y Group Organizer.

- **POST user/register/** Utilizada en Vista: Registro, Registra un nuevo usuario en el sistema, Crea nuevas instancias en las tablas: User, Competitor y Organizer.
- **POST user/request-reset-email/** Utilizada en Vista: Contraseña Olvidada, Envía un correo para recuperar contraseña a la dirección de correo electrónico indicada.
- **POST user/password-reset/id/token/** Utilizada en Vista: Contraseña Olvidada. Permite acceder a la vista para introducir una nueva contraseña usando el id del usuario y el token temporal de este.
- **PATCH user/password-reset-complete/** Utilizada en Vista: Contraseña Olvidada. Si se introduce una nueva contraseña se cambia en el sistema la contraseña del usuario.
- **POST user/logout/blacklist/** Utilizada en Vista: Home, barra de navegación. Permite la desconexión del usuario en el sistema. Agregando los tokens de Access y Refresh a una lista negra para que no se puedan utilizar nuevamente.
- **POST user/personal/** Utilizada en Vista: Home, barra de navegación. Obtiene los datos personales de un usuario: Nombre, Apellido e institución.
- **POST user/editpersonal/** Utilizada en Vista: Home, barra de navegación. Permite editar los datos personales de un usuario: Nombre, Apellido e institución.

Endpoints del Módulo Competition:

Endpoints que interactúan con datos sobre las competencias, obtienen o modifican información de las tablas: Track, Benchmark, Acronyms, Submission y Results.

- **GET competition/filter/organize/pk/** Utilizada en Vista: Home. Obtiene todas las instancias de competencias en las que el usuario conectado en el sistema es organizador.
- **GET competition/filter/participate/pk/** Utilizada en Vista: Home. Obtiene todas las instancias de competencias en las que el usuario conectado en el sistema está participando.
- **GET competition/** Utilizada en Home, en Search A track. Obtiene todas las instancias de competencias en el sistema.
- **POST, PUT competition/newTrack/** Utilizada en Vista: Creación de track. Crea una nueva competencia, también permite editarla.
- **GET competition/trackBenchmark/pk/** Utilizada en Vista: Track. Obtiene toda la información necesitada en la vista track, obteniendo todos los valores de una competencia en particular de la tabla Track, Benchmark, MetricSelection, GraphicSelection y otros datos específicos como número de competidores.

- **POST competition/participate/pk/** Utilizada en Vista: Track. Si un usuario ejecuta esta petición participará en la competencia que elija, si ejecuta nuevamente la petición saldrá de la competencia.
- **POST competition/submission/** Utilizada en Vista: Track. Permite a un usuario hacer una entrega en una competencia.
- **GET competition/submissionUser/pk/** Utilizada en Vista: Track. Obtiene un listado de las entregas de un usuario en una competencia, permite mostrarle al usuario el historial de entregas.
- **GET competition/resultsLeaderboard/** Utilizada en Vista: Track. Obtiene la información necesaria para mostrar el leaderboard, obteniendo la métrica con la que se ordena y las filas de la tabla.
- **GET competition/resultsByCompetitor/** Utilizada en Vista: Track. Obtiene los resultados de un usuario si es que este es competidor. El usuario verá los resultados de todas sus entregas según los gráficos que el administrador haya permitido que se vean.
- **GET competition/resultsAdministrator/** Utilizada en Vista: Track. Obtiene los resultados de un usuario si es que este es administrador. El usuario puede ver todos los resultados de todas las entregas y usuarios de la competencia.
- **GET competition/acronyms/pk/** Utilizada en Vista: Track. Obtiene una lista de los acrónimos utilizados por los administradores para representar las entregas de los usuarios.
- **GET competition/downloadFilePrivateMedia/pk/** Utilizada en Vista: Track. Utilizada por el usuario para descargar entregas del historial, solo el usuario que hizo la entrega y el organizador del track tienen acceso a estos archivos.
- **GET competition/downloadSubmissions/pk/** Utilizada en Vista: Track. Descarga todas las entregas de todos los usuarios que participan en la competencia, esta petición solo la puede realizar un organizador de la competencia.

4.5. Inicio de sesión

El sistema de inicio de sesión está implementado usando JWT Tokens usando la librería simple-jwt [22]. Al usuario se le entregan tokens que debe enviar al servidor en cada petición, el servidor verifica que los tokens sean correctos y no hayan expirado, si no hay problemas entonces el servidor entrega la vista y la información pedida por el usuario. Los tokens son JSON encriptados que contiene datos que identifican al usuario e información del algoritmo utilizado en la encriptación.

Se decidió utilizar este tipo de inicio de sesión dado que se le quita responsabilidad al servidor. Este no debe recordar si el usuario ha hecho inicio de sesión o no, sino que son los tokens que se guardan en el equipo del usuario los que cumplen esta función. Para más detalle, el sistema de inicio de sesión funciona con dos tokens: Access Token y Refresh token. El Access token se envía con cada petición, permitiendo obtener las vistas de la aplicación si

es que este es correcto y no ha expirado, y el Refresh Token tiene la función de obtener un nuevo Access Token si es que expira.

4.6. Evaluación de entregas de usuarios

La petición **POST competition/submission/** utiliza el método *evaluate()* para evaluar las entregas de los usuarios y retornar resultados. El método toma como entrada los archivos de tipo .cla del Groundtruth (que representan el dataset), la matriz de distancia del usuario y las métricas elegidas por el organizador.

Por cada uno de los archivos de clasificación se obtiene la información de estos guardándola en diferentes variables. Se obtienen dos arreglos que definen cuales son las clases del Target y Query y también se construyen dos arreglos que definen a que clase pertenece cada objeto de los dos archivos. Finalmente se obtiene la cantidad de objetos total en cada archivo.

A continuación, se procede a evaluar la entrega del usuario entregando un diccionario con los resultados almacenados. Por cada una de las filas de la matriz de distancia (donde cada una representa a un objeto distinto en el Query) se evalúan las métricas escogidas por el organizador. Para realizar esto se recorre cada fila almacenando las distancias en un arreglo y luego se ordenan de forma ascendente, es decir, la primera distancia del arreglo es la que tiene el menor valor de la fila. Luego cada distancia se transforma en un 1 o un 0 dependiendo si el objeto del Target respectivo a esa distancia era de la misma clase que el del objeto que representa a la fila, de esta forma obtenemos un arreglo ordenado de objetos relevantes, donde los 1 representan a los objetos relevantes y un 0 a los objetos que no.

Cuando se obtiene el arreglo ordenado de objetos relevantes se procede a recorrerlo para evaluar las métricas elegidas por el organizador, obteniendo los resultados para cada métrica y guardándolos en un diccionario. Mientras se recorre este arreglo se construye una tabla que almacena los valores de precision y recall.

Posteriormente se promedian los resultados de las métricas utilizando todas las filas de la matriz de distancia, sumando los resultados de todas las filas y dividiéndolos por el número total de objetos en el Query. Además, en este proceso, utilizando las clases de las filas y la clase del “Nearest neighbor” se calculan los valores utilizados en la matriz de confusión de la aplicación.

Finalmente, estos datos obtenidos que representan los resultados de las métricas, los valores de precision-recall y los valores de la matriz de confusión, se guardan en un JSON retornado por el método *evaluate()*. Los valores son posteriormente guardados en una fila de la tabla Resultados, donde también se define el usuario que entregó la matriz, el track y el JSON.

4.7. Gráficos del sistema

En la pestaña Results de las páginas de las competencias se pueden observar diferentes gráficos que muestran el rendimiento de los algoritmos. Los gráficos utilizados son un gráfico de líneas y un heatmap para mostrar la información de Precision-Recall y una matriz de confusión respectivamente. Para implementar estos gráficos se hizo uso Highcharts, esta es una librería creada con JavaScript y TypeScript que entrega todas las herramientas para crear gráficos que permiten visualizar información.

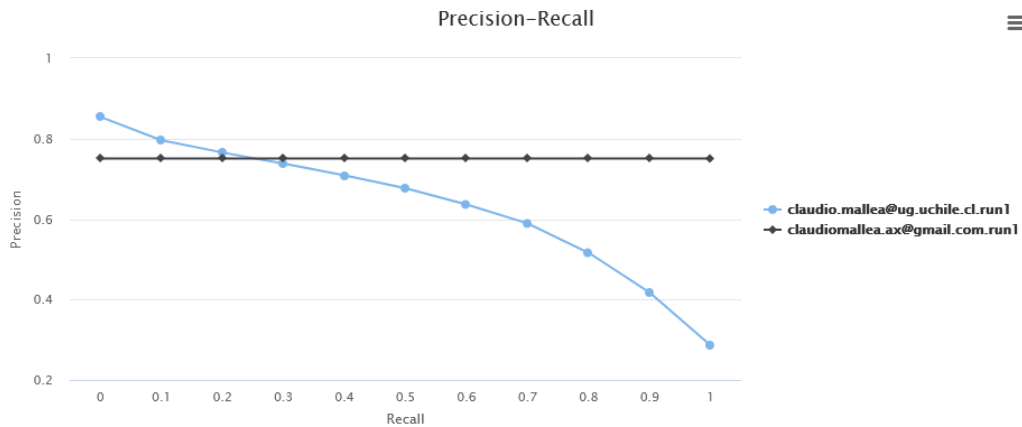


Figura 4.6: Ejemplo gráfico tipo “Precision-Recall”.

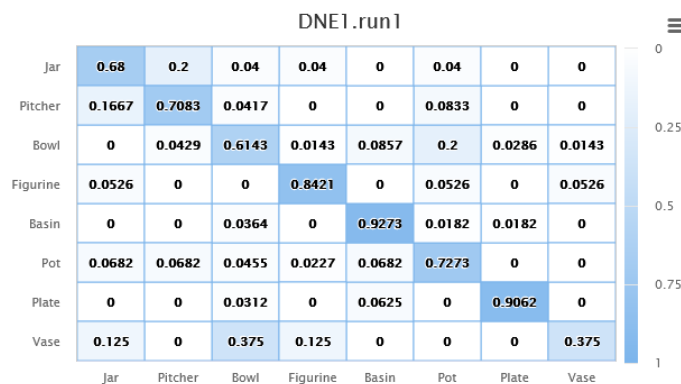


Figura 4.7: Ejemplo gráfico tipo “Matriz de confusión”.

En las figuras 4.6 y 4.7 se pueden observar los gráficos de Precision-Recall y Matriz de Confusión respectivamente. El gráfico de Precision-Recall muestra líneas que representan cómo se comporta el valor de precisión de un algoritmo con respecto a un recall del 0 al 1. Si se apunta el cursor a la línea, se informará sobre el valor más detallado de precision en esa posición, indicando también sobre el nombre de la entrega del usuario, las demás líneas se hacen transparentes para observar mejor la línea apuntada. A la derecha también se encontrarán los nombres de las entregas de los usuarios, indicando el color de la línea respectiva, por defecto las entregas se identifican con el nombre del usuario y el número del

intento con el que se envió. Con respecto a la matriz de confusión, se muestran las secciones cuadrículadas donde en su interior se encuentra un valor con respecto a la columna y fila. A la derecha se muestra una barra de color en degrade, indicando que los colores más transparentes representan a valores cercanos a cero y al contrario a los valores más cercanos al 1.

Los gráficos de Highcharts presentan un botón en la parte superior derecha donde se pueden encontrar diferentes funcionalidades. En este se puede expandir el gráfico en pantalla completa, se puede imprimir el gráfico y se puede exportar a 4 posibles tipos de archivo: PNG, JPEG, PDF y SVG. De esta manera el organizador tiene la posibilidad de exportar gráficos de manera individual, sin necesidad de descargar todos los gráficos al mismo tiempo.

Para realizar la exportación de gráficos a PDF y SVG. Highcharts presenta dos opciones, utilizar el servidor de exportación que Highcharts mantiene en sus servidores o levantar este mismo servicio, que es un servidor creado con Nodejs, en la máquina del desarrollador. No es posible utilizar el servidor de exportación online de Highcharts debido a que no se permiten más de 10 peticiones por minuto, y teniendo en cuenta que cada gráfico es una petición y por cada competidor se utilizan hasta 4 gráficos, entonces se sobrepasaría el límite rápidamente. Por esta razón se decide utilizar el servidor creado con NodeJs para levantarlo en la misma máquina de AWS utilizado para levantar la aplicación.

4.8. Deployment

La aplicación web se encuentra alojada en un servidor de Amazon Web Services. En un principio se pensó alojar el sistema en un servidor de Heroku [23], debido a que este no presenta costos, pero este servicio tiene límites los que podrían perjudicar el funcionamiento del sistema. Los límites fueron la baja capacidad de almacenamiento y la cualidad de que las máquinas de Heroku se demoran en reaccionar a peticiones si es que estas no se han hecho en plazos de tiempo muy extensos entre sí. Finalmente se tomó la decisión de alojar la aplicación en un servidor de AWS debido a que el estudiante tenía experiencia con este servicio.

Como se mencionó anteriormente, se decidió utilizar Docker en la aplicación. Para hacer esto se necesita escribir un archivo requirements.txt con todas las librerías utilizadas que se instalarán y un archivo .yml que define todos los contenedores que se crearán. Estos contenedores son 4: El Back-End donde también se define que se utilizará Gunicorn [24], un proxy de Nginx [25], un contenedor para utilizar Let's Encrypt [26] y un contenedor que contiene el servidor de Highcharts. El archivo .yml también menciona todos los “volúmenes” que se utilizarán, estos son los espacios de almacenamiento que no se eliminan si es que un contenedor deja de funcionar. Esto sirve para que no se pierdan los datos de los usuarios si es que algo le pasa al servidor.

Posteriormente al uso de Docker, se compró un nombre de dominio. En un principio se pensó en utilizar el nombre de dominio que entrega AWS por defecto en sus máquinas, pero teniendo en cuenta que la página será utilizada por muchos organizadores y competidores que tendrán que insertar el nombre para acceder a este servicio, se estimó por comprar un nombre de dominio para facilitar el acceso. El nombre del dominio es “www.retrievaltracks.com” debido a que representa a una página donde se encuentran competencias de problemas de

tipo recuperación (o *retrieval*). El dominio fue comprado en BlueHost [27] y se utiliza el servicio Route53 de AWS para conectar esta URL con la IP del servidor.

Por último, la aplicación hasta esta etapa del deployment utilizaba *HTTP* para sus peticiones, pero debido a que el sistema tiene información privada que puede ser accedida debido al uso de este protocolo, fue necesario hacer que la aplicación utilizara *HTTPS*. Para realizar esto se deben obtener certificados TLS de una autoridad certificada. Debido a que se utilizó Docker para subir el sistema, solo fue necesario agregar un nuevo contenedor al archivo `.yml` de Docker llamado `Acme-companion` [28] que obtiene certificados de Let's Encrypt automáticamente.

Capítulo 5

Validación

En este capítulo se describirá el proceso de validación de la aplicación web. Se describirán las entrevistas realizadas y los procesos que se realizaron para medir la efectividad de la solución. Además, se reporta la experiencia y comentarios entregados por los usuarios que participaron en el proceso.

Como se determinó en la concepción de la solución, la validación se realizó por medio de entrevistas y encuestas para medir la usabilidad y utilidad del sistema, además de otros procesos para la validación de funcionalidades de la aplicación. Como este sistema tiene dos tipos de usuarios (organizadores y competidores) fue necesario comunicarse con diferentes personas de estos dos grupos para tener entrevistas de 40 minutos aproximadamente. El estudiante se comunicó por correo electrónico con organizadores de SHREC preguntando si accedían a participar en estas reuniones de los cuales accedieron 5 organizadores. Con respecto al grupo de competidores, se comunicó con 10 estudiantes a los cuales se les explicó el contexto de la aplicación. En un principio se pensó que se necesitaría gente que entendiera íntegramente algunos conceptos referentes a problemas de tipo *retrieval*, pero esto no fue necesario, debido que para realizar cualquier acción en la aplicación web no se necesita entender estos conceptos, si es que ya se tienen los archivos necesarios que se deben subir.

5.1. Datos utilizados

Para realizar la evaluación del sistema fue necesaria la obtención de archivos y datos para la creación de una competencia, además de posibles entregas que pueda hacer un participante en estas. Los archivos utilizados fueron los entregados por el profesor guía, estos corresponden a archivos utilizados y obtenidos en una competencia de SHREC organizada por el profesor, la competencia corresponde a SHREC 2021: Retrieval of Cultural Heritage Objects [29]. Estos archivos corresponden a los utilizados en el primer desafío de la competencia, refiriéndose a los dos archivos de Target y Query utilizados en el Groundtruth. Con respecto a los archivos para realizar entregas, se obtuvieron 6 matrices de distancias que corresponden a algoritmos diferentes. Los resultados de estos algoritmos para las métricas: *Mean Average Precision*, *First-Tier*, *Second-Tier*, *Nearest Neighbor* y *Normalized Discounted Cumulative Gain* eran

conocidos, además de las matrices de confusión y los gráficos de Precision-Recall.

5.2. Validación de funcionalidad

Para definir que las funcionalidades de la aplicación estaban correctamente implementadas entregando los resultados deseados, se decidió realizar diferentes procesos. En primer lugar, utilizando un framework para evaluar REST APIs llamado Insomnia [30], se revisaron todos los endpoints de la API del sistema. Probando con inputs correctos e incorrectos, para determinar que la respuesta (refiriéndose al mensaje y estado recibido de las peticiones) fueran los correctos. En este proceso se identificaron las peticiones con problemas para poder solucionarlos.

Una vez se solucionaron las peticiones con errores, se procedió a verificar que cuando una petición fallara cuando un usuario estuviera utilizando la aplicación se mostrara un mensaje adecuado indicando este problema. Los problemas pueden ser de tres tipos: Validación del input en un formulario de la aplicación, errores de permisos y errores del servidor.

Cuando un usuario está utilizando un formulario en el sistema y este procede a entregar un dato que no tiene el formato pedido, la aplicación debe entregar un aviso de este error, informándole al usuario sobre el formato correcto. Para revisar esto se revisó cada input de cada formulario en el sistema, probando con formatos incorrectos e identificando el aviso correspondiente. Estos procesos se realizan en el Front-End, si de alguna forma el usuario lograra evitar la interfaz y entregar una petición con formatos incorrectos, entonces será interceptado con un error del Back-End donde también se validan estos inputs. Estas validaciones que realiza el Back-End fueron evaluadas al momento de recorrer la API en el proceso anteriormente mencionado.

Con respecto a errores de permisos, donde solo un tipo de usuario (Competidor o Organizador) puede acceder a la información deseada en el sistema, el Back-End se encarga de estos errores, validando el tipo de usuario. Para revisar esto se crearon dos tipos de usuarios en Insomnia y se intentó acceder a las diferentes peticiones de la API donde el tipo de usuario fuera relevante. En este proceso no se encontró ningún error, dado que el sistema entrega una página con estado 403 Forbidden, en cada petición donde el usuario sea el incorrecto.

Con respecto a errores del servidor, si el servidor no pudiera entregar la información pedida por el usuario, entonces se muestra un aviso en la página indicando la petición e indicando que el error corresponde a un problema del servidor. Para revisar esto se modificó cada petición de la API haciendo que entregaran errores y luego se recorrieron todas las interfaces del sistema, identificando si los avisos estuvieran siendo expuestos.

Luego de identificar los errores anteriores y haberlos solucionado, se procedió a verificar cada funcionalidad de la aplicación a excepción de la evaluación de entregas de usuarios, recorriendo las posibles funcionalidades que permite el sistema. Para realizar esto se crearon 2 cuentas y se procedió a crear y participar en competencias creadas por estos usuarios, verificando que las competencias aparecieran en el menú Home, que fuera posible la creación de competencias y la participación de estas, el cambio de contraseña y los cambios de la confi-

guración de las competencias. Se identificó que todas las funcionalidades fueron operativas a excepción de algunos errores que fueron solucionados posteriormente.

Con respecto a la evaluación de entregas de usuarios, se hizo uso de las 6 matrices de distancia y de los archivos de groundtruth dispuestos para la validación del sistema. Se crearon 7 usuarios, donde uno fuera un organizador y creara una competencia con los archivos de groundtruth mencionados anteriormente y los demás usuarios fueron participantes de esta competencia, donde cada uno entregara una matriz de distancia distinta. Los resultados de esta evaluación eran los esperados, donde los resultados de las métricas, y los gráficos de Precision-Recall y matrices de confusión coincidieron con los mismos que se tenían antes de la creación de la aplicación.

5.3. Medición de utilidad

Para medir la utilidad de la página se definieron tareas que hacer en el sistema para los usuarios. Estas tareas se hicieron con el objetivo de verificar que los usuarios pudieran disponer de las funcionalidades del sistema de manera efectiva, revisando si es que estas son intuitivas, que no generaran confusión y si es que se presentaban errores o problemas. Se definieron tareas diferentes para los dos grupos de usuarios. A continuación, se listarán las tareas respectivas a cada grupo.

5.3.1. Tareas organizadores

- Crea una cuenta y hacer inicio de sesión en la página.
- Crea una nueva competencia.
- Vuelve al home y cambia la configuración de usuario, cambiando tu institución.
- Vuelve a tu competencia.
- Cambia el nombre de tu track y agrega un nuevo contacto de organizador.
- Revisa el leaderboard del track.
- Cambia las métricas que se mostrarán en el leaderboard.
- Descarga el leaderboard en formato CSV.
- Revisa el gráfico de: General Precision-Recall en los resultados.
- Revisa cual es el gráfico específico de Precision-Recall del usuario: claudiomallea.a@gmail.com.
- Cambia cuáles gráficos pueden ver tus usuarios en la sección de Resultados.
- Descarga todos los gráficos de tus usuarios en PDF.

5.3.2. Tareas competidores

- Crea una cuenta e ingresa a la pantalla de home de la aplicación.
- Busca el track “Testing Track” que está en proceso y participar en él.
- Debes hacer una entrega de tu algoritmo.
- Descarga el archivo subido.
- Verifica en qué posición quedó tu algoritmo en el Leaderboard.
- Verifica el gráfico de precisión recall de tu resultado.
- Haz una nueva subida de archivo.
- Verifica si el resultado es mejor que el anterior.
- Revisa la configuración de usuario.

5.3.3. Resultados del proceso

Las entrevistas fueron realizadas sin problemas y de manera efectiva. El proceso consistió en una explicación del contexto de la aplicación en el inicio de la reunión, luego se entregaron los archivos que se utilizarían en el proceso, posteriormente se realizaron las tareas junto al usuario y finalmente se dio un espacio para realizar comentarios. Las entrevistas fueron realizadas mediante Zoom y en inglés, debido a que los participantes tenían diferentes nacionalidades, pero todos hablaban este idioma.

Con respecto a los participantes pertenecientes al grupo de organizadores, la mayoría de las tareas fueron realizadas de manera correcta y efectiva. Se estimó un tiempo máximo de un minuto por cada tarea, pero los usuarios lograron hacerlas antes de este tiempo de forma rápida. En las tareas 7 y 11 existieron problemas debido a que algunos participantes no pudieron realizar lo que se pedía teniendo que pasar a las siguientes tareas. Estas dos tareas tienen que ver con los permisos que un organizador tiene para editar algunas configuraciones del track, y cuando se les pedía a los usuarios cambiar algunas configuraciones del sistema no sabían en qué parte de la interfaz podían hacer esto.

Durante el proceso de la realización de tareas se encontraron dos errores del sistema, donde los usuarios entendían lo que se les pedía y trataban de hacerlo, pero el sistema no estaba respondiendo. Uno de estos errores se encontró en la primera entrevista realizada y consistía en que en la tarea 7 el usuario cambiaba la métrica del leaderboard en la sección correspondiente pero el sistema no lo cambiaba. El otro error fue encontrado mientras se resolvía el anterior, donde se encontró que al editar la configuración del track las fechas de inicio y fin cambiaban a la fecha actual. Estos errores fueron corregidos posterior a la validación.

Con respecto a los participantes que actuaron de competidores en el sistema, todas las tareas se realizaron correctamente. No existieron casos donde los competidores no pudieran hacer las tareas indicadas. Sin embargo, algunas tareas tomaron más tiempo de lo estimado.

En la tarea 5 a los participantes les costó encontrar su usuario en el Leaderboard, esto debido a que se utiliza un seudónimo en las tablas, luego de un tiempo lograron encontrar su usuario. En esta misma tarea a los participantes se les dificultó encontrar la métrica con la que se ordena el leaderboard.

5.4. Medición de usabilidad

Luego de que finalizará una entrevista con algún participante, se le indicó si podía responder una encuesta que fue enviada a su correo. Para realizar estas encuestas se utilizó el System Usability Scale (SUS) [31] esta es una herramienta para medir la usabilidad de los sistemas, este consiste en 10 preguntas con 5 posibles respuestas, desde “Totalmente en desacuerdo” a “Totalmente de acuerdo”. A continuación, se listarán las preguntas que son utilizadas en este sistema.

1. Creo que me gustaría usar esta aplicación con frecuencia.
2. Encontré esta aplicación innecesariamente compleja.
3. Creo que la aplicación es fácil de usar.
4. Creo que necesitaría ayuda para utilizar este sistema.
5. Encontré que las funcionalidades del sistema están bien integradas.
6. Pienso que hay mucha inconsistencia en la aplicación.
7. Me imagino que la mayoría de la gente aprendería a usar esta aplicación de forma rápida.
8. Encontré esta aplicación difícil de utilizar.
9. Me sentí cómodo usando esta aplicación.
10. Necesité aprender muchas cosas antes de poder usar la aplicación.

Con respecto a las preguntas 7 y 10, se les informó a los usuarios que aprender los conocimientos para desarrollar algoritmos o los tecnicismos que se usan en la competencia no contaban para estas preguntas.

5.4.1. Resultados del proceso

La encuesta fue respondida durante los meses de diciembre y enero, donde se recibieron 10 respuestas, como se mencionó anteriormente estas personas consistían en algunos de los entrevistados.

Esta metodología de medición de usabilidad indica ciertas reglas para obtener el puntaje del sistema, por cada usuario se obtiene un resultado y luego se saca el promedio para obtener una nota. A continuación, se listarán los pasos para obtener estos resultados.

1. Convertir la escala de Totalmente en desacuerdo a Totalmente de acuerdo a una escala numérica del 1 al 5.
2. Calcular $X =$ Suma de los puntos para preguntas con número impar
3. Calcular $Y =$ Suma de los puntos para preguntas con número par
4. Calcular $X_0 = X - 5$
5. Calcular $Y_0 = 25 - Y$
6. Calcular SUS Score = $(X_0 + Y_0) * 2,5$

Luego de hacer este proceso por cada uno de los usuarios y obtener un promedio se obtiene un puntaje de 92.75 lo que es equivalente a una nota A o Excelente en la escala establecida por el sistema. Esto nos indica que las interfaces del sistema están cumpliendo correctamente sus objetivos al tener una alta usabilidad. La escala utilizaba para obtener la nota, además del formulario enviado a los participantes se puede ver en el Apéndice B del anexo.

5.5. Comentarios

Durante las entrevistas y en un espacio al final de estas los participantes fueron entregando comentarios sobre el sistema, opinando sobre que les parecía. Los comentarios fueron muy positivos indicando que les gustó la aplicación, la encontraron intuitiva, fácil de usar y útil. Los organizadores de SHREC indicaron que este sistema les iba a servir bastante debido a que todos los problemas que enfrentan en la creación de competencias se podían realizar de forma rápida y automática (refiriéndose a la evaluación de algoritmos) en la aplicación.

Los organizadores comentaron que en algunas competencias que ellos realizan se necesitan algunas métricas que son muy específicas para el problema que necesitan resolver. Sin embargo, también comentaron que están constantemente realizando competencias que utilizan solamente las métricas que tiene la aplicación, por lo que comentaron que definitivamente iban a utilizar la página.

Adicionalmente se les preguntó sobre que funcionalidades podrían faltar en el sistema. A continuación, se entrega un listado de los comentarios. Cabe destacar que los comentarios estaban en inglés originalmente, pero se han traducido al español.

- “Por ahora no tengo ningún comentario crítico que entregar, me gustó mucho la aplicación en general, ¡sigan con el buen trabajo!”
- ”Tal vez sea muy temprano, pero tal vez una página FAQ podría ser útil, probablemente venga naturalmente después de la primera fase de testing“
- “No estoy seguro como podría ser posible expandir con respecto a las métricas de rendimiento. Algunos organizadores, en el pasado requirieron métricas adicionales como las ROC curves. Probablemente sea útil prever algún modulo para expandir las métricas propuestas (que sean pertinentes y estándar) ”

- “Mi principal preocupación es la flexibilidad de las evaluaciones. Por ejemplo, la competencia que yo organizo es mayoritariamente sobre *classification* y *retrieval*, y necesitaría más que métricas de *retrieval*. Sería bueno si pudiéramos subir nosotros el script que será ejecutado por cada entrega, pero entiendo que podría ser mucho trabajo.”

Al observar los comentarios podemos darnos cuenta de algunas preocupaciones de los usuarios con respecto a la flexibilidad de la aplicación. Estas preocupaciones están relacionadas a los límites que se impuso en el proyecto para obtener una aplicación completa y funcional en los tiempos establecidos. Con respecto a las métricas que algunos usuarios puedan necesitar debido a la especificidad de sus competencias, el sistema es flexible debido a que para ingresar una nueva métrica se debe ingresar los datos que identifican a esta en la tabla Metrics y agregar el código de esta en el script de evaluación de entregas de usuarios, siendo sencillo el proceso para realizar esto. Con respecto a las competencias con problemas de tipo *classification*, la forma de abordar estos problemas en el futuro se explicará en el siguiente capítulo.

Capítulo 6

Conclusión

El objetivo principal de este trabajo de título consistía en acelerar, facilitar y optimizar el proceso de evaluación de algoritmos de recuperación de contenido 3D por medio de la creación de una aplicación web, donde se espera que los usuarios reciban los resultados de la evaluación de forma rápida y automática, sin la necesidad de la intervención del grupo organizador a cargo del benchmark. Este objetivo, junto a los objetivos específicos para llevarlo a cabo, se cumplieron con éxito, los organizadores actualmente pueden acceder a la aplicación y crear sus propias competencias de manera rápida y las evaluaciones de los algoritmos de los usuarios se realizan automáticamente, dejando el código de las métricas en un repositorio público para que exista transparencia en la evaluación.

El trabajo tuvo una etapa de validación al final, donde el estudiante validó la funcionalidad del sistema y midió la utilidad y usabilidad de este. Con este propósito, se hizo uso de archivos de groundtruth y matrices de distancia obtenidas de una competencia de SHREC, donde ya se sabían los resultados de estos. Para validar la funcionalidad se realizaron diferentes procesos, se verificó el correcto funcionamiento de la API, se recorrió cada funcionalidad de las interfaces de la aplicación verificando que funcionarían con el propósito con el que estaban diseñadas y se hizo uso de los archivos obtenidos para recrear la competencia de SHREC de forma de verificar que los resultados dados por las entregas de los algoritmos sean los mismos de los que ya se conocían. Los errores encontrados en la validación de funcionalidad fueron corregidos en su totalidad.

Para medir la usabilidad y utilidad del sistema, el estudiante se comunicó con diferentes personas, algunas de las cuales fueron organizadores de competencias en SHREC. Se realizaron entrevistas con ellos, de forma que hicieran tareas en la aplicación y luego se les indicó responder una encuesta con preguntas sobre esta, dando espacio para entregar sus comentarios. Los resultados de este proceso fueron positivos, las funcionalidades de la aplicación funcionaban como se esperaba y las interfaces eran lo suficientemente intuitivas para que usuarios puedan realizar la mayoría de las tareas sin problemas. Además, se obtuvieron buenos comentarios por parte de los organizadores de SHREC, mostrándose optimistas con la aplicación e indicando que estaban expectantes por utilizarla.

Con respecto a la metodología utilizada para la realización de este trabajo, el estudiante considera que esta fue correcta. Al tener reuniones semanales con el profesor guía, el estu-

dante obtuvo un feedback continuo sobre la aplicación y el desarrollo de esta. Además, se considera que la división del trabajo en etapas de: obtención de requisitos, diseño, desarrollo y evaluación del proyecto, fue un proceso necesario y efectivo para un trabajo de software de este tipo, dado que se pudo concluir con cada etapa de forma efectiva obteniendo una aplicación que cumple con los objetivos planteados.

Con respecto a las principales herramientas utilizadas para el desarrollo de la aplicación, se concluye que fueron las correctas para este trabajo. La utilización de Django en el Back-End, al ser una herramienta con la que el estudiante está familiarizado, debido al uso que se la ha dado en proyecto de semestres anteriores, resultó en un desarrollo continuo e intuitivo, con leves problemas al utilizar librerías desconocidas, pero que de todas formas fueron utilizadas correctamente. El uso de React en el Front-End también fue una decisión correcta. por las mismas razones anteriores, además de entregar una velocidad en la obtención de interfaces por partes del usuario que no se observan en páginas que utilizan otras herramientas. El uso de PostgreSQL también se considera una decisión acertada, debido a que se le puede dar uso a este gestor de base de datos por medio de las funcionalidades que entrega Django y las funciones o formas de interactuar que utiliza este framework con cualquier gestor ya eran conocidas para el estudiante, por lo que no fue necesario aprender demasiados conceptos o tecnicismos para hacer uso de este. Los problemas que se generaron en el uso de otras herramientas serán abordados en la siguiente sección.

6.1. Lecciones aprendidas

La realización de este proyecto resultó en muchos conocimientos nuevos para el estudiante, teniendo que entender y utilizar herramientas de las que se tenía poca experiencia, refiriéndose a herramientas como Docker, algunas librerías utilizadas en Django y todas las aplicaciones utilizadas para el deployment del sistema. Algunas partes de este proceso fueron complejas, estresantes y se transformaron en un verdadero desafío, debido a que se habían definido plazos de desarrollo donde no se había tenido en cuenta el tiempo que toma la curva de aprendizaje cuando se utilizan nuevas herramientas. Este problema resultó en iteraciones no consideradas en el desarrollo que tomaron mucho más tiempo del esperado. Esta experiencia es enriquecedora para el estudiante debido a que en futuros proyectos del mismo nivel debe tener en consideración la utilización de tiempo y/o períodos de estudio, investigación y aplicación de nuevas herramientas, las cuales deben considerarse en la planificación inicial.

Otra lección que se aprendió en el proceso fue sobre la importancia del uso de herramientas para organizar y ubicar adecuadamente las tareas diarias y semanales. En un inicio el estudiante desarrollaba partes del sistema sin algún registro de los requisitos que se estaban implementando, solo valiéndose de los comentarios que se realizaban en GitHub. Afortunadamente se tomó en cuenta este problema y se comenzó a utilizar a tiempo la herramienta Trello [32] lo que le permitió obtener una mayor organización en actividades, tareas e hitos del proyecto. Si el estudiante pudiera realizar este trabajo de memoria otra vez, se utilizaría una aplicación o software que apoyara eficientemente las labores de planificación y organización de este tipo de proyectos.

6.2. Posibles trabajos futuros

Este proyecto da la oportunidad de la realización de muchos trabajos futuros. Esto se debe a los límites impuestos en los objetivos del trabajo para poder alcanzar el desarrollo de todas las funcionalidades en el tiempo establecido. Como se mencionó en el capítulo 3, uno de los límites de este proyecto es la forma en que los competidores entregan sus algoritmos. El único formato que se dispone es el de uso de matrices de distancia, pero en algunas competencias de SHREC los competidores utilizan otros formatos de entrega de algoritmos, por lo que este proyecto se podría mejorar abarcando a un público usuario más amplio si es que se implementan otros tipos de formatos de entrega.

Otro tipo de mejora al sistema es la creación de nuevas métricas y gráficos. En este proyecto se implementaron las métricas y gráficos más utilizados por los organizadores en SHREC, pero no se han implementado algunos algoritmos que sirven para tareas que son más específicas como: e-measure, tier images, curvas Roc, Distancia de Hausdorff, Coeficiente de Sorensen-Dice y el índice de Jaccard (estas métricas fueron nombradas por los organizadores de SHREC en el proceso de obtención de requisitos). Actualmente, debido a la naturaleza de las métricas implementadas, los organizadores no tendrían un motivo para no seleccionar todas las métricas que ofrece el sistema, pero se esperaría que en un futuro donde si se hayan habilitado nuevas métricas que los organizadores puedan seleccionar y configurar métricas que solo ellos necesiten.

Por la forma en que está implementado el sistema, nos podemos dar cuenta que un organizador de un problema que no sea necesariamente de recuperación en contenido 3D puede utilizar perfectamente esta aplicación, si es que va a utilizar los formatos indicados por el sistema. De esta forma se podrían crear competencias de recuperación en contenido que no sea necesariamente de objetos 3D, sino que otro tipo de contenido. El único problema es que el sistema dispone de un formato de entrega de ground-truth que es específico de la recuperación en contenido 3D. Pero si en un futuro se agregan más formatos para el ground-truth y el proceso de evaluación se adapta a estos formatos para otro tipo de problemas, entonces la aplicación podría ser útil para estos también, por lo que se tiene una posible oportunidad para la realización de un trabajo en el futuro.

Por último, este sistema solo permite competencias en donde se tengan que resolver problemas del tipo *retrieval*, pero por la forma en que fue desarrollada la interfaz, se podría permitir otros tipos de problemas, para esto se podría añadir una sección en la creación de competencias, donde se pueda seleccionar el tipo de problema, especificando métricas que solo se utilizan en *retrieval* o *classification*, además de tener esto en cuenta en la forma en que se entregan los algoritmos de los usuarios.

Bibliografía

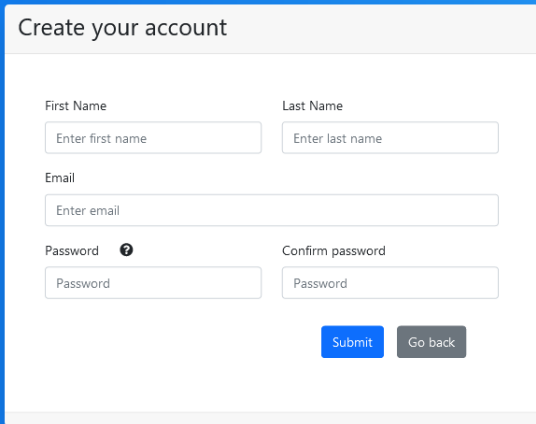
- [1] *Shape Retrieval Challenge*. <https://www.shrec.net/>, visitado el 2021-05-04.
- [2] *SHREC 2020: Retrieval of digital surfaces with similar geometric reliefs*. http://shrec.ge.imati.cnr.it/shrec20_pattern_retr/index.html, visitado el 2021-05-04.
- [3] *SHREC 2020: Track on multi-domain protein shape retrieval*. <http://shrec2020.drugdesign.fr/>, visitado el 2021-05-04.
- [4] *Workshop on 3D Object Retrieval*. <https://3dor2021.github.io/>, visitado el 2021-05-04.
- [5] *Completion 3D Benchmark*. <https://completion3d.stanford.edu/>, visitado el 2021-05-04.
- [6] *Codabench: What is Codalab?* <https://worksheets.codalab.org/>, visitado el 2021-05-04.
- [7] *Kaggle*. <https://www.kaggle.com/>, visitado el 2021-05-04.
- [8] *Amazon Web Services*. <https://aws.amazon.com/es/>, visitado el 2022-02-16.
- [9] *React*. <https://reactjs.org/>, visitado el 2022-02-16.
- [10] *Django*. <https://www.djangoproject.com/>, visitado el 2022-02-16.
- [11] *PostgreSQL*. <https://www.postgresql.org/>, visitado el 2022-02-16.
- [12] *Highcharts*. <https://www.highcharts.com/>, visitado el 2022-02-07.
- [13] *Elastic Computing 2*. <https://aws.amazon.com/es/ec2/>, visitado el 2022-02-16.
- [14] *Amazon Linux 2*. <https://aws.amazon.com/es/amazon-linux-2/>, visitado el 2022-02-24.
- [15] *Amazon Elastic Block Store*. <https://aws.amazon.com/es/ebs/>, visitado el 2022-03-23.
- [16] *Balsamiq*. <https://balsamiq.com/>, visitado el 2022-02-24.
- [17] *Django REST Framework*. <https://www.django-rest-framework.org/>, visitado el 2022-02-16.

- [18] *React-Bootstrap*. <https://react-bootstrap.github.io/>, visitado el 2022-02-16.
- [19] *Docker*. <https://www.docker.com/>, visitado el 2022-02-16.
- [20] *GitHub*. <https://github.com/>, visitado el 2022-02-16.
- [21] *The Princeton Shape Benchmark Classification Format*. https://shape.cs.princeton.edu/benchmark/documentation/classification_format.html, visitado el 2022-02-18.
- [22] *Simple-Jwt*. <https://django-rest-framework-simplejwt.readthedocs.io/en/latest/>, visitado el 2022-02-16.
- [23] *Heroku*. <https://www.heroku.com/>, visitado el 2022-02-16.
- [24] *Gunicorn*. <https://gunicorn.org/>, visitado el 2022-02-24.
- [25] *Nginx*. <https://www.nginx.com/>, visitado el 2022-02-24.
- [26] *Let's Encrypt*. <https://letsencrypt.org/>, visitado el 2022-02-24.
- [27] *Bluehost*. <https://www.bluehost.com/>, visitado el 2022-02-21.
- [28] *Acme-Companion*. <https://github.com/nginx-proxy/acme-companion>, visitado el 2022-02-21.
- [29] *SHREC 2021: Retrieval of Cultural Heritage Objects*. <http://www.ivan-sipiran.com/shrec2021.html>, visitado el 2022-03-23.
- [30] *Insomnia*. <https://insomnia.rest/>, visitado el 2022-03-23.
- [31] *System Usability Scale*. <https://www.usability.gov/how-to-and-tools/methods/system-usability-scale.html>, visitado el 2022-02-16.
- [32] *Trello*. <https://trello.com/>, visitado el 2022-02-18.

Anexos

Anexo A

Otras interfaces del sistema



The image shows a registration form titled "Create your account" centered on a blue background. The form is white with a light gray border and contains the following fields and buttons:

- First Name:** A text input field with the placeholder text "Enter first name".
- Last Name:** A text input field with the placeholder text "Enter last name".
- Email:** A text input field with the placeholder text "Enter email".
- Password:** A text input field with the placeholder text "Password" and a small question mark icon to its right.
- Confirm password:** A text input field with the placeholder text "Password".
- Submit:** A blue button with white text.
- Go back:** A gray button with white text.

Figura A.1: Vista de registro de nueva cuenta en la aplicación.

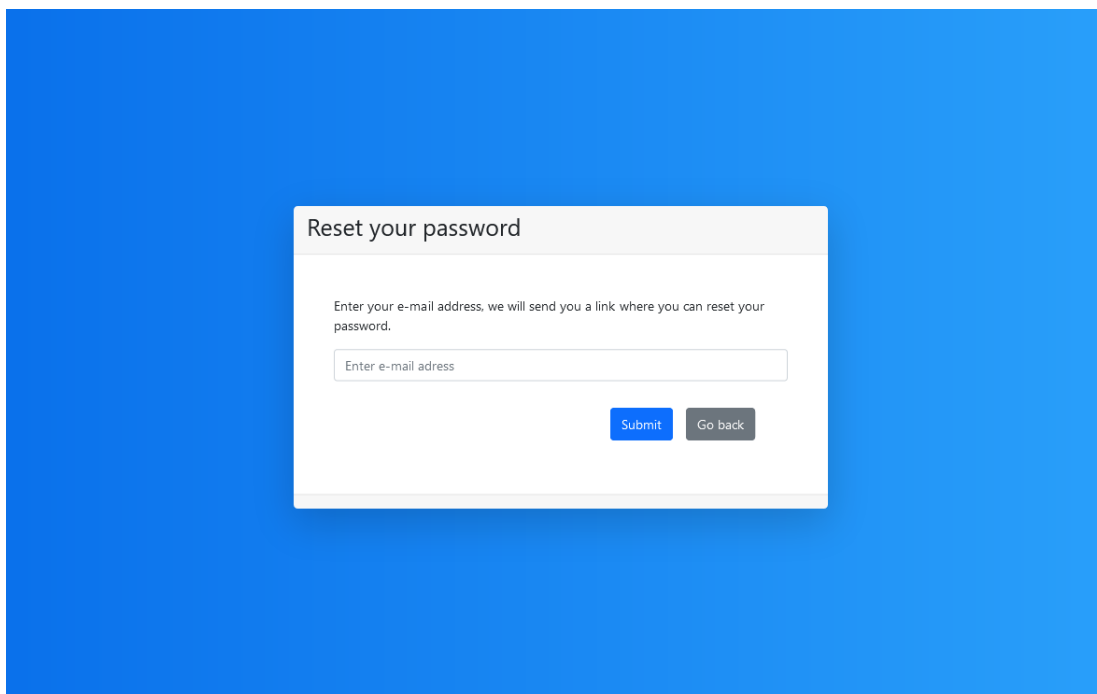


Figura A.2: Vista para recuperación de contraseña. Se indica ingresar el correo del usuario.

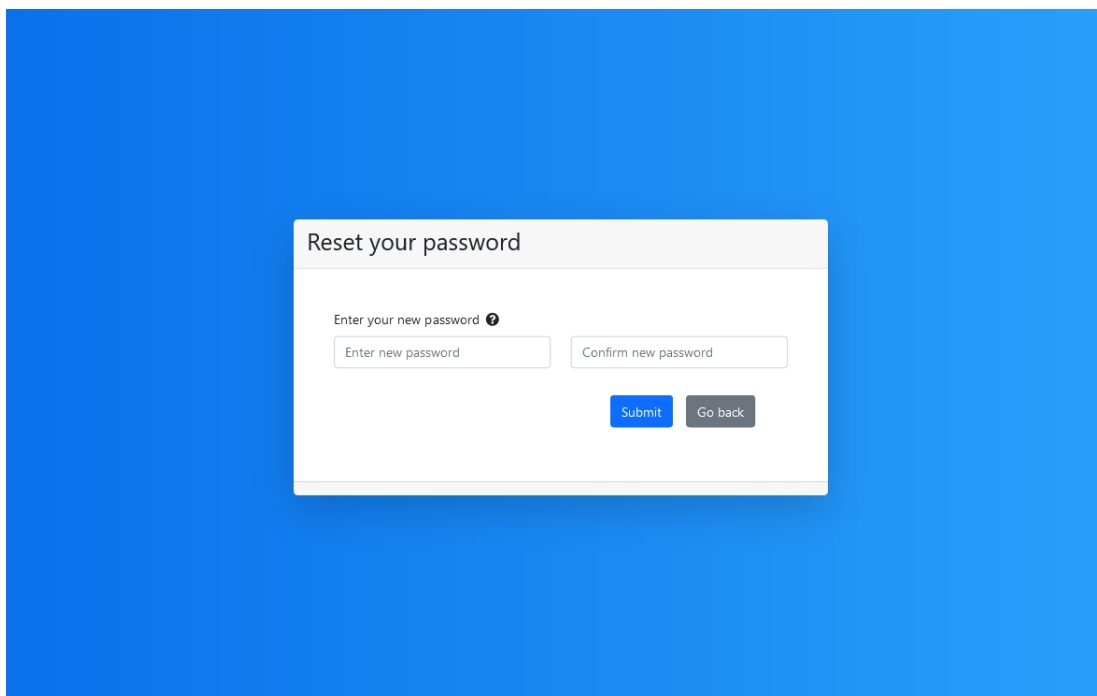


Figura A.3: Vista para recuperación de contraseña. Se indica ingresar la nueva contraseña.

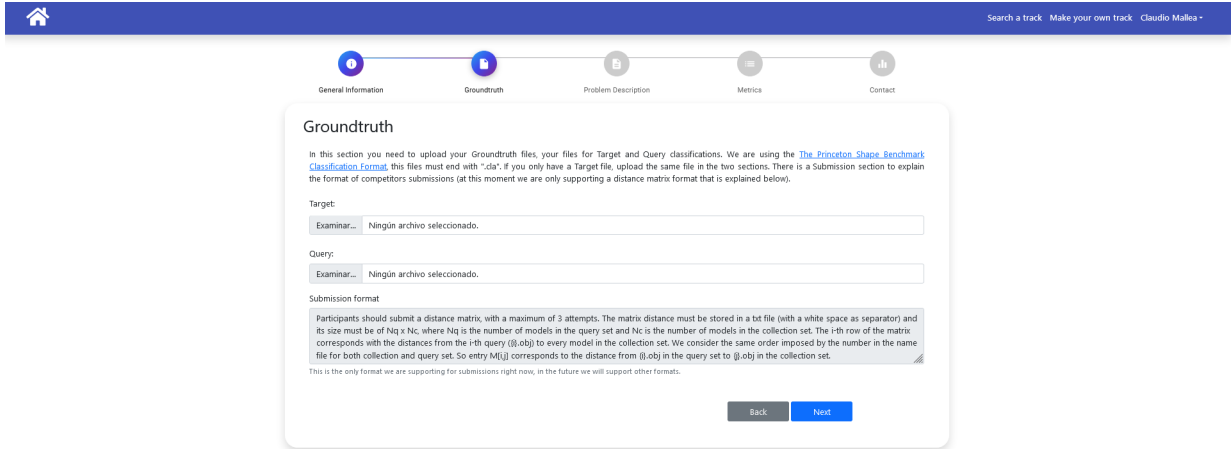


Figura A.4: Sección Groundtruth en vista: Creación de track.

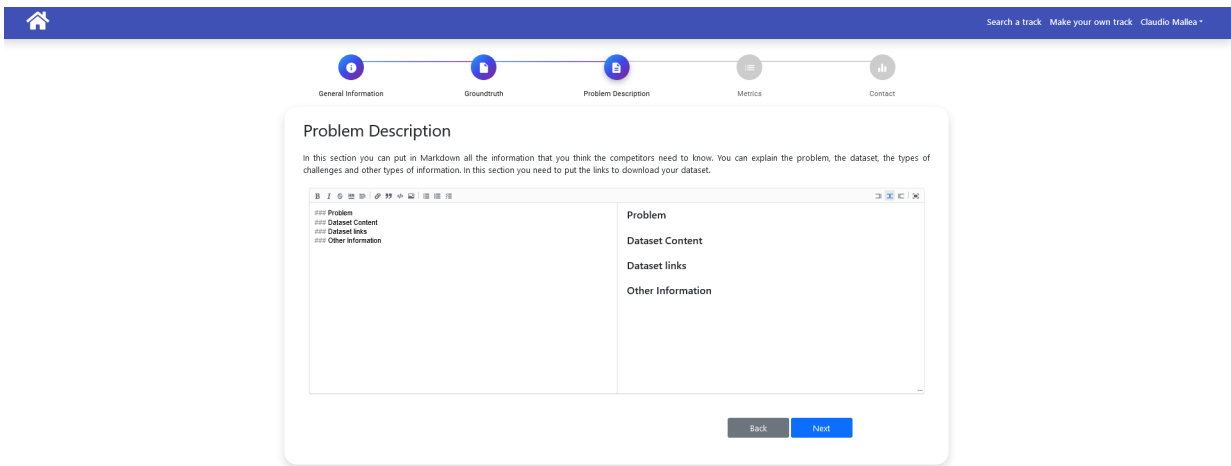


Figura A.5: Sección Problem Description en vista: Creación de track.

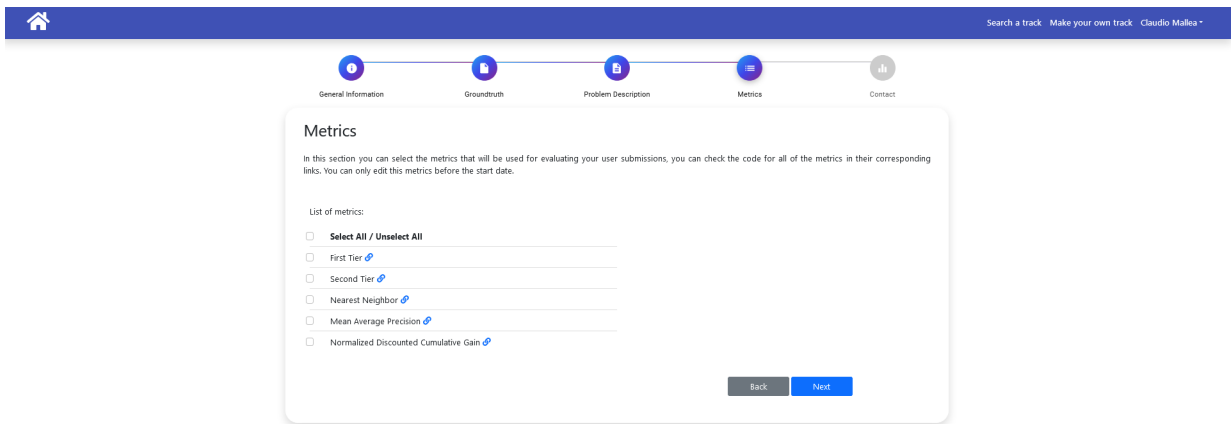


Figura A.6: Sección Metrics en vista: Creación de track.

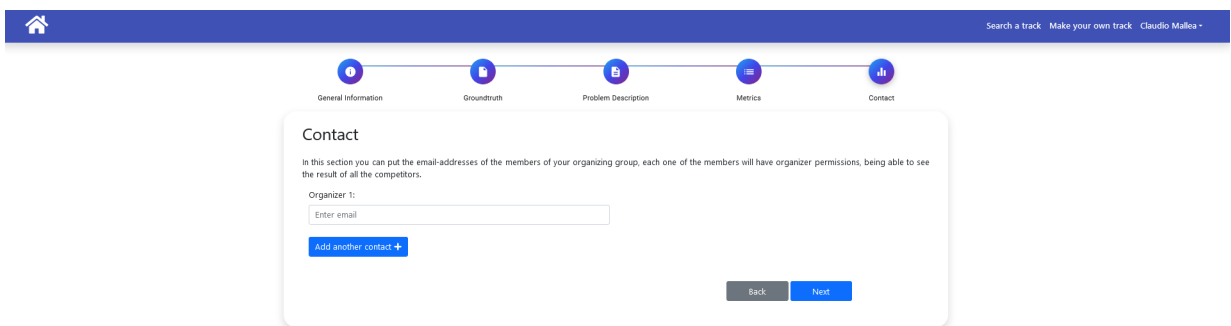


Figura A.7: Sección Contacts en vista: Creación de track.

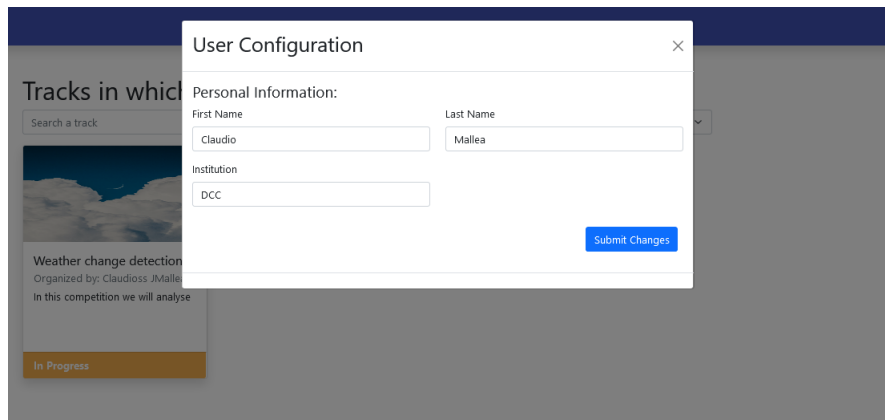


Figura A.8: Ventana emergente para cambiar configuración de usuario.

General Information | **Leaderboard** | Results | Submissions | Contact

Leaderboard

As an administrator, you can change what metrics will be displayed in the leaderboard, you can change this in the Leaderboard Settings, you can also change the metric that will be used for sorting the leaderboard.

This table shows all the results of all the metrics, sorted by the metric: **ST**

[Export to CSV](#) [Export to TeX](#)

#	User	FT	ST ▼	NN	MAP	NDCG
1	c@c.com	0.6388	0.7358	0.5993	0.6915	0.8428
2	u@u.com	0.6388	0.7358	0.5993	0.6915	0.8428
3	p@p.com	0.6388	0.7358	0.5993	0.6915	0.8428
4	r@r.com	0.6494	0.5162	0.7256	0.678	0.8755
5	g@g.com	0.6494	0.5162	0.7256	0.678	0.8755
6	j@j.com	0.6494	0.5162	0.7256	0.678	0.8755
7	a@a.com	0.5813	0.4385	0.7509	0.6273	0.8704
8	f@f.com	0.5813	0.4385	0.7509	0.6273	0.8704

Figura A.9: Pestaña Leaderboard en vista: Track.

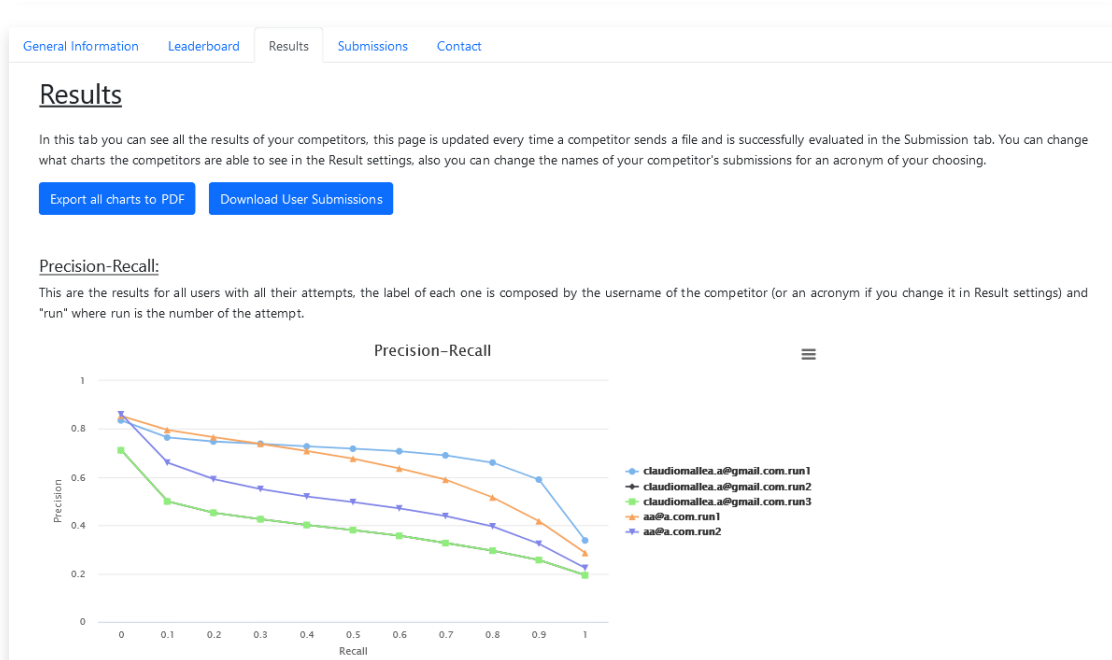


Figura A.10: Pestaña Results en vista: Track, accedida por Organizador. Se muestran los botones para exportar gráficos y el gráfico de Precision-Recall de todas las entregas.

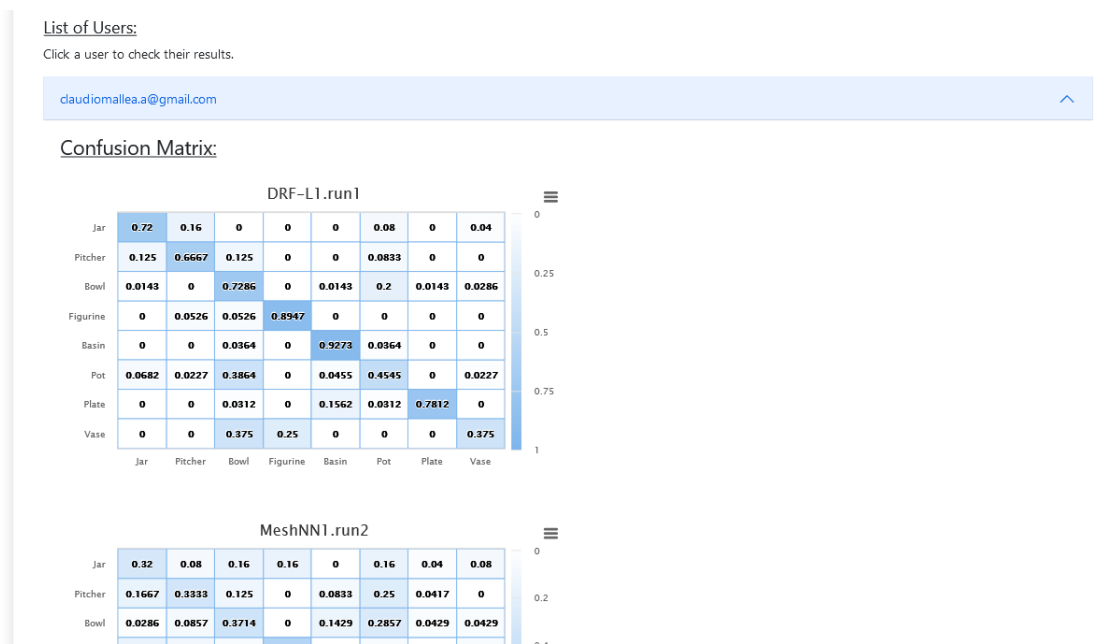


Figura A.11: Pestaña Results en vista: Track, accedida por Organizador. Listado donde el organizador puede acceder en detalle a los resultados de un competidor.

Results

This are the results of your submissions, you can check in detail how different submissions compare to each other. For each submission, the first part of the name is the name of the file of your submission, the second part (after the dot) is the number of attempt.

Export all charts to PDF

Best results:

This are the results of your best submission (best FT) , you got this results in your **first attempt**.

Confusion Matrix:

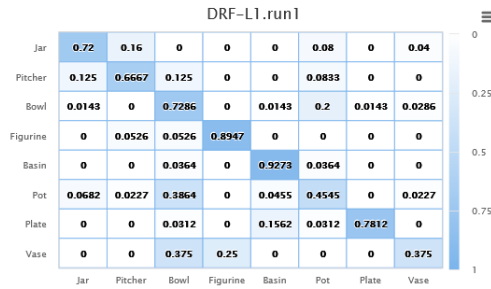
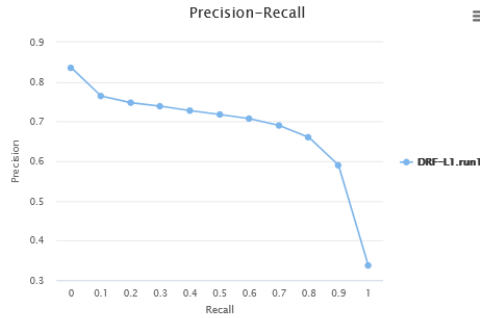


Figura A.12: Pestaña Results en vista: Track, accedida por Competidor.

Precision-recall:



All results:

This are the results of all your attempts.

Confusion Matrix:

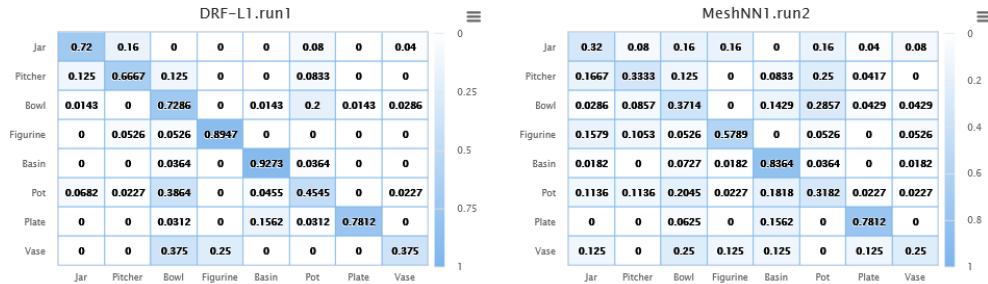


Figura A.13: Pestaña Results en vista: Track, accedida por Competidor.

General Information Leaderboard Results Submissions Contact

Submissions

Start Date: 2022-01-26

End Date: 2022-01-29

Maximum number of attempts: 3

Description:

Participants should submit a distance matrix, with a maximum of 3 attempts. The matrix distance must be stored in a txt file (with a white space as separator) and its size must be of $N_q \times N_c$, where N_q is the number of models in the query set and N_c is the number of models in the collection set. The i -th row of the matrix corresponds with the distances from the i -th query ($(i).obj$) to every model in the collection set. We consider the same order imposed by the number in the name file for both collection and query set. So entry $M[i,j]$ corresponds to the distance from $(i).obj$ in the query set to $(j).obj$ in the collection set.

Evaluation metrics:

- First Tier [↗](#)
- Second Tier [↗](#)
- Nearest Neighbor [↗](#)
- Mean Average Precision [↗](#)
- Normalized Discounted Cumulative Gain [↗](#)

Submit your file:

Submit [History](#)

Examinar... Ningún archivo seleccionado. [Submit](#) There is 0 attempts left

If your submission fails, you dont lose an attempt.

Figura A.14: Pestaña Submissions en vista: Track.

General Information Leaderboard Results Submissions Contact

Contact information

The administrator of this track is: Claudio Mallea

Members of the organizing group:

#	Role	E-mail address
1	Administrator	claudiomallea.a@gmail.com
2	Organizer	claudio.mallea@ug.uchile.d

Figura A.15: Pestaña Contacts en vista: Track.

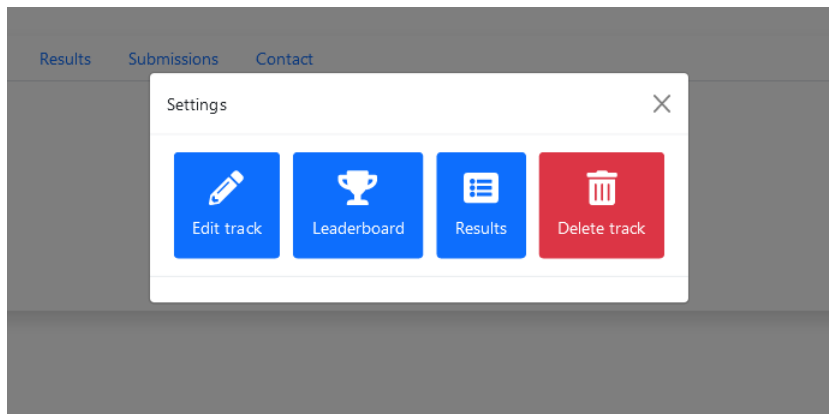


Figura A.16: Settings de la vista: Track, muestra las cuatro posibles opciones de configuración en la competencia.

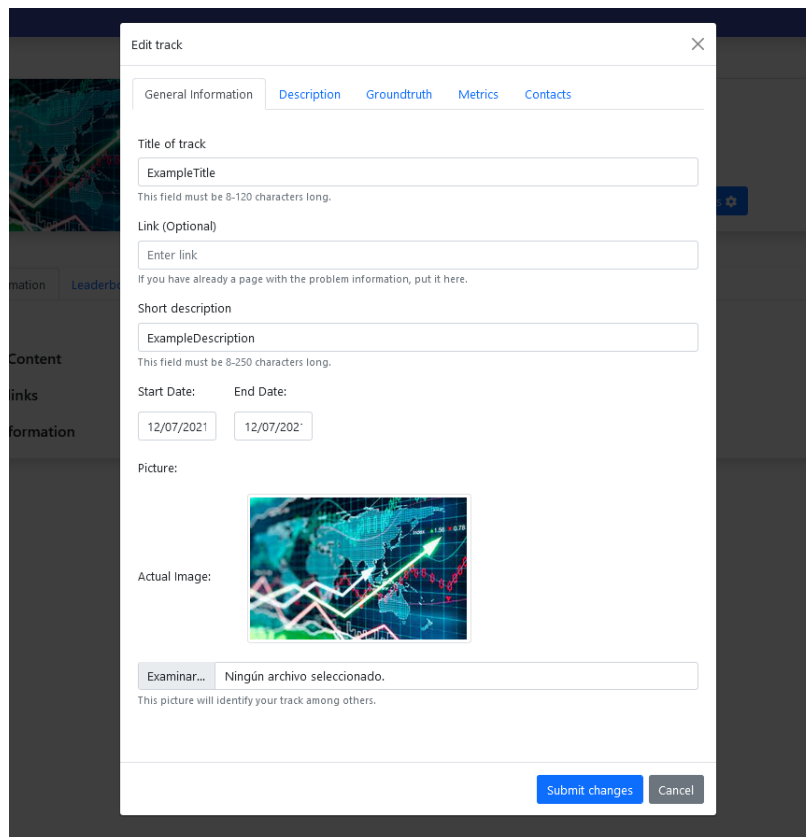


Figura A.17: Opción Edit Track de Settings.

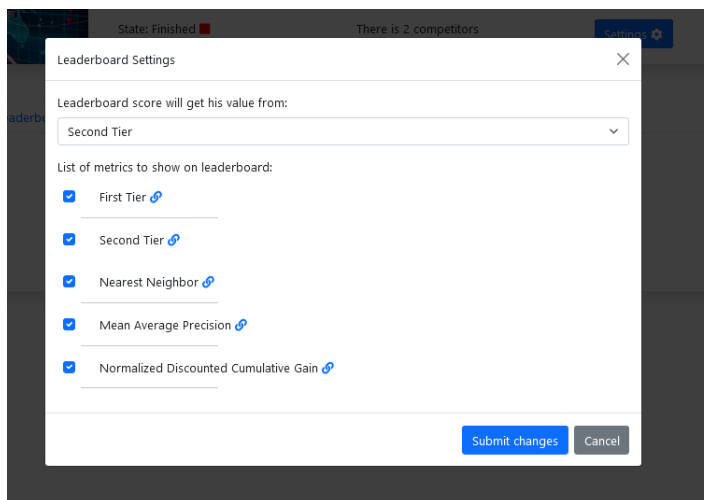


Figura A.18: Opción Leaderboard Settings de Settings.

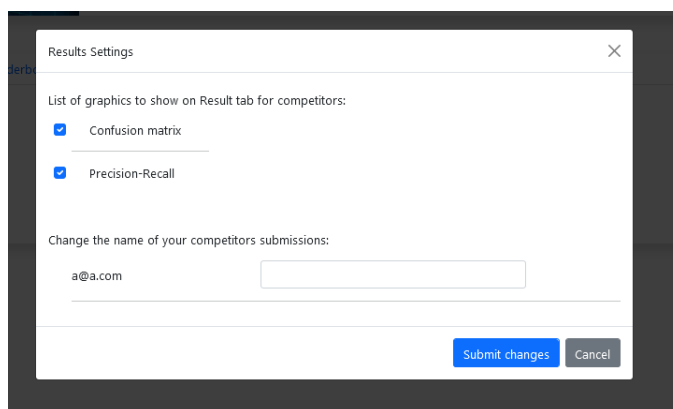


Figura A.19: Opción Result Settings de Settings.

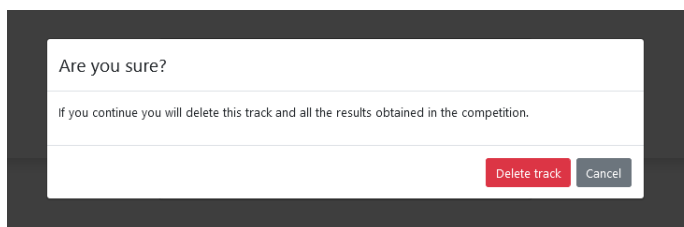


Figura A.20: Opción Delete track de Settings.

Anexo B

System Usability Score

The image shows a web-based usability form titled "Retrieval tracks Usability form". The form is set against a light purple background. On the right side, there is a vertical toolbar with icons for zooming, printing, and other navigation functions. The form content is as follows:

Retrieval tracks Usability form
Thank you for being part of the validation stage of the Retrieval tracks application. Next there is a form with 10 questions where you have to select an option from 1 to 5.

Correo electrónico *
Correo electrónico válido
Este formulario recopila correos electrónicos. [Cambiar la configuración](#)

I think that I would like to use this system frequently *
Strongly disagree 1 2 3 4 5 Strongly agree

I found the system unnecessarily complex *
Strongly disagree 1 2 3 4 5 Strongly agree

I thought the system was easy to use. *
Strongly disagree 1 2 3 4 5 Strongly agree

I think that I would need the support of a technical person to be able to use this system. *
Strongly disagree 1 2 3 4 5 Strongly agree

I found the various functions in this system were well integrated. *
Strongly disagree 1 2 3 4 5 Strongly agree

Figura B.1: Formulario enviado a participantes de validación, primeras 5 preguntas.

I thought there was too much inconsistency in this system. *

1 2 3 4 5

Strongly disagree Strongly agree

I would imagine that most people would learn to use this system very quickly. (Context: people who already know or have been in retrieval tracks contests) *

1 2 3 4 5

Strongly disagree Strongly agree

I found the system very cumbersome to use. *

1 2 3 4 5

Strongly disagree Strongly agree

I felt very confident using the system. *

1 2 3 4 5

Strongly disagree Strongly agree

I needed to learn a lot of things before I could get going with this system. *

1 2 3 4 5

Strongly disagree Strongly agree

Figura B.2: Formulario enviado a participantes de validación, últimas 5 preguntas.

SUS Score	Grade	Adjective Rating
> 80.3	A	Excellent
68 – 80.3	B	Good
68	C	Okay
51 – 68	D	Poor
< 51	F	Awful

Figura B.3: Escala de notas al utilizar SUS.

Anexo C

SHREC - 3D Shape Retrieval Challenge

SHREC2022 - 3D Shape Retrieval Challenge 2022

<http://www.shrec.net/>

At Eurographics 2022 Symposium on 3D Object Retrieval, 1-2 September 2022

New since 2020: Full paper submissions will follow a two-stage review process and will be published in the international journal *Computes & Graphics* upon acceptance.

We strongly encourage to consider the graphics replicability stamp initiative, <http://www.replicabilitystamp.org/>, and to apply for this additional sign of recognition.

Introduction

The general objective of the 3D Shape Retrieval Challenge is to evaluate the effectiveness of 3D-shape retrieval algorithms. SHREC2022 is the sixteenth edition of the challenge. Like previous years, it is organized in conjunction with the Eurographics Symposium on 3D Object Retrieval, where the results will be reviewed and presented at the symposium.

Thanks to the efforts of previous track organizers, SHREC already provides many resources to compare and evaluate 3D retrieval methods. For this year's contest, we aim to explore new and updated tracks. Therefore, the participants are invited to have an active role in the organization of the event. This includes proposing track themes, building or acquiring a test collection, and deciding upon the queries, relevance assessment, and performance measures.

The participants of each track will collectively write a paper, which will be peer reviewed, and published in *Computes & Graphics* upon acceptance. At least one author per track must register for the symposium, and present the results. We also cordially invite all participants of a track to register and attend the workshop.

Tracks

The following tracks are organized. For description of tasks, the collections, queries, the evaluation procedure, and time schedule, follow the links or contact the track organizer.

1. Online detection of heterogeneous gestures
Organizers: Marco Emporio, Anton Pirtac, Ariel Caputo, Marco Cristani, Andrea Giachetti (VIPS lab, University of Verona)
Contact: [Andrea Giachetti](#)
[Web page](#)
2. Fitting and recognition of simple geometric primitives on point clouds
Organizers: Chiara Romanengo, Andrea Raffo (CNR-IMATI, Italy)
Contact: [Chiara Romanengo](#)
[Web page](#)
3. Open-Set 3D Object Retrieval using Multi-Modal Representation
Organizers: Yue Gao, Yifan Feng, Xibin Zhao (Tsinghua University, China), Yandong Guo (OPPO Inc.)
Contact: [Yifan Feng](#)
[Web page](#)
4. Pothole and crack detection on road pavement using RGB-D images
Organizers: A. Ranieri, E. Moscoso Thompson, S. Biasotti (CNR-IMATI, Italy)
Contact: [Elia Moscoso Thompson](#)
[Web page](#)
5. Sketch-Based 3D Shape Retrieval in the Wild
Organizers: Jie Qin (Nanjing University of Aeronautics and Astronautics, China), Shuaihang Yuan (New York University, USA), Jiaxin Chen (Beihang University, China), Boulbaba Ben Amor (IMT Nord Europe, France), Yi Fang (NYU Abu Dhabi, UAE)
Contact: [Jie Qin](#)
[Web page](#)
6. Protein-ligand binding site recognition
Organizers: L. Gagliardi, W. Rocchia (IIT, Italy), A. Raffo, U. Fugacci, S. Biasotti (CNR-IMATI, Italy)
Contact: [Andrea Raffo](#)
[Web page](#)

Figura C.1: Pantalla inicial actual de SHREC, las competencias se muestran en el listado de la sección inferior de la figura.

SHREC 2021: Retrieval of Cultural Heritage Objects

The Track

3D shape analysis has a historical symbiosis with the cultural heritage domain. The list of contributions of computer graphics and graphics technology in cultural heritage applications is long, and it includes 3D digitization, reconstruction, virtual restoration, immersive interfaces, and so on. Most of these techniques rely on having a suitable 3D representation of archaeological objects, making it possible to process and analyze these objects with computational methods effectively. Nevertheless, some contributions of shape analysis and geometry processing in CH applications have been limited by the relatively small number of items we can use in experimentation.

This track presents our initiative to promote the research of 3D shape analysis methods in CH domains. Our dataset contains digitized versions of archaeological objects from pre-Columbian cultures in Peru, with varied geometry and artistic styles. The real artifacts are in the Josefina Ramos de Cox (JRC) museum in Lima, Perú. The models were scanned as part of a research project funded by the National Agency for Science and Technology in Peru (CONCYTEC) and in conjunction with the Pontifical Catholic University of Peru, which oversees the administration of the mentioned museum.

In this track, we present two retrieval challenges considering two aspects: the shape and the culture. Regarding the shape, archaeologists in the JRC museum classified the scanned objects by shape using specific taxonomies for Peruvian pre-Colombian artifacts. Regarding culture, the JRC museum keeps records of the pre-Colombian cultures to which the artifact belongs. We collected all this metadata for the scanned models, which serves as input for our retrieval tasks.

The proposed challenges have different degrees of complexity. Retrieval by shape is probably the more affordable challenge given that there exist suitable methods in the 3D shape retrieval literature to deal with geometric characterization. Nevertheless, there are cases where the distinction between objects in different classes is barely perceived. On the other hand, retrieval by culture is a more difficult challenge. Models from the same culture can have varied shapes, and probably the most distinguishable characteristic is the combination of geometry and painting style.

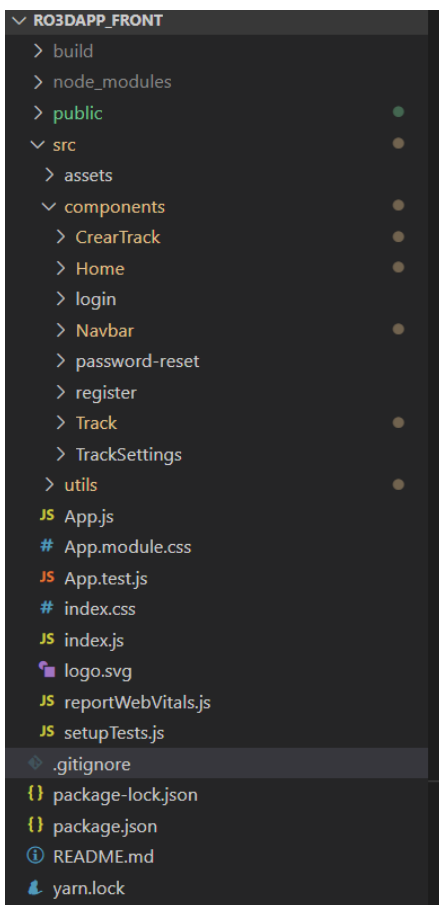
The dataset

The dataset consists of 3D scanned models from cultural heritage objects captured in the Josefina Ramos de Cox museum in Lima, Perú. The technology used to acquire the 3D models was a structured-light desktop scanner, which produces high-resolution 3D textured models. We applied a post-processing step to normalize the position by translating the objects' center to the origin of 3D space. We also change the orientation manually, such that shapes are oriented up in the Y-axis. We keep the original scale of models because the scale can be a distinctive feature that differentiates objects. Finally, we simplify the triangular mesh of each shape to have nearly 40,000 triangle faces.

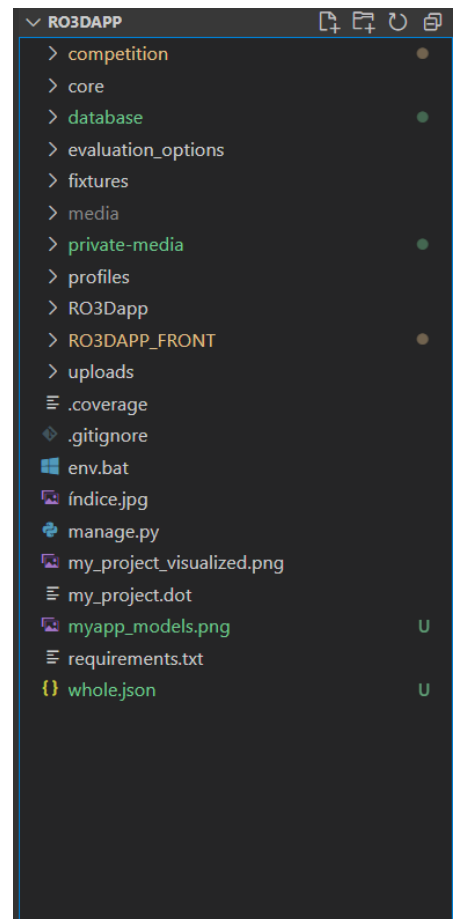
Figura C.2: Ejemplo de competencia actual en SHREC.

Anexo D

Estructuras del código



(a) Front-End



(b) Back-End

Figura D.1: En la figura (a) se muestra la estructura de carpetas del Front-End mostrando los componentes y en la figura (b) se muestra la estructura para el Back-End.