



# Desarrollo de una red social de arte sonoro y visual a partir de datos de personas usuarias.

Alejandro Acosta

Licenciatura en Música y Tecnología. Escuela Universitaria de Artes. Universidad Nacional de Quilmes

En este artículo planteamos el desarrollo de una red social de arte sonoro y visual. Por medio de esta se busca que las personas usuarias puedan generar archivos a partir de sus propios datos y compartirlos mediante una aplicación móvil. Esta plataforma nos puede ser útil para visibilizar una problemática actual que posee dos ejes que son la privacidad y la vida cotidiana en el ámbito virtual. Por una lado, intentaremos contribuir a evidenciar el conflicto ético de la utilización de nuestros datos por distintas entidades; por otro, abriremos una línea de investigación partiendo desde el supuesto de que los datos que generamos cotidianamente contienen un potencial narrativo y que este potencial puede ser utilizado como material compositivo. De este modo, podremos desarrollar una app que se sostiene a través del almacenamiento y control de datos por parte de sus usuaries. Así como analizar la sustancialidad de dichos datos como materia compositiva, principalmente para crear música mediante la generación de archivos. En un contexto virtual donde intervienen entidades comerciales o políticas, se naturaliza la apropiación de nuestras experiencias, se desdibuja el límite de los consentimientos y se manipula la identidad, allí encontramos un campo propicio para la exploración artística. Mediante esta aplicación, buscamos visibilizar dicha problemática actual.

Palabras claves: *Música, datos, aplicación.*

## 1.INTRODUCCIÓN

El presente trabajo fue desarrollado en el marco de la Beca de formación en la Investigación para Estudiantes en la etapa final de Carrera de Grado (BEFI 2020) otorgada por la Secretaría de Investigación de la Universidad Nacional de Quilmes. Este trabajo fue planteado en dos etapas: Una etapa inicial que se basa en el desarrollo del software en un entorno API Rest de código abierto. Y una segunda etapa de investigación a partir de las preguntas: “¿Los datos de la vida cotidiana digital poseen potencial narrativo y pueden estos ser utilizados como material compositivo?”. A partir de la experiencia de la persona usuaria podremos proceder a obtener datos relevantes tanto para intentar responder a estas preguntas como para visibilizar y analizar la problemática de la privacidad de nuestros datos en la actualidad..

Consideramos que esta propuesta se encuentra en la convergencia entre el discurso artístico y el tecnológico y que tiene lugar en la red con un perfil que nos parece se suma al término *Net.sound*

*art*[1] porque es una red social soportada en una aplicación móvil que será utilizada para crear piezas de arte sonoro de manera online, es así que

recibe de internet su forma, su constante expansión y su significado.

Existe una corriente de pensadores que analizan el problema de la intimidad de las personas en las redes[2][3][4][5] así como del valor que tienen hoy sus datos generados[6][7][8][9]. También las hay referidas a esta temática varias obras que habitan el ciberespacio, ya sea funcionando aun o no[10][11], así como obras que refieren a este tema en formato físico[12][13]. Por otro lado, encontramos varias obras que indagan y reflexionan acerca de la narrativa que poseen los datos de la vida cotidiana[6] y la analogía que se puede trazar obra y vida cotidiana a través de los datos.

En el proyecto que abordamos intervienen música e imagen a través de tecnologías que, si bien son

actuales, poseen un proceso evolutivo y un devenir histórico y cultural que les permiten ser tales en la actualidad. Desde la analogía que presentan la música escrita y la ejecutada, desde los primeros registros fonográficos que cambiaron el paradigma de la escucha hasta la irrupción del arte concreto y el electrónico y la resignificación del soporte y la fuente sonora. Desde el fonógrafo hasta el registro de audio en un teléfono móvil para ser subido a la nube, la humanidad ha intentado retener y manipular el sonido y lo ha hecho siempre almacenando datos en algún tipo de soporte y mediante la tecnología existente. Sin meternos en las profundidades que guarda el acontecimiento histórico del registro fonográfico para la cultura, podemos tomarlo para este trabajo como un antecesor directo simplemente porque es el primer almacén de datos musicales audible. De aquí en adelante artistas sonoros como Russolo, la invención de objetos y su *Il risveglio di una città* (1913); Schaeffer y la música concreta registrada todavía en disco en *Étude aux chemins de fer* (1948), Cage y sus ediciones multipista de procesos aleatorios en *Williams Mix* (1952) o Lucier con *I'm sitting in a room* (1969), entre tantos otros, irán a la par de las tecnologías emergentes en sus tiempos, ya sean mecánicas, electrónicas o digitales.

Para el almacenamiento de datos de registro, datos generados a través de la app y archivos creados decidimos usar bases de datos no relacionales (NoSQL)<sup>1</sup> porque poseen las ventajas de almacenar información sin una estructura predefinida (clave-valor, documentos, gráficos, columnas); Están preparadas para escalar de forma horizontal (añadiendo más nodos al sistema); son capaces de ofrecer grandes volúmenes de datos de forma mucho más rápida de la que lo harían los tradicionales modelos relacionales. Para trabajar con estas bases de datos optamos por el sistema MongoDB. Este es un motor de base de datos NoSQL basado en documentos, es decir, que guarda los datos en estructuras parecidas a un Json<sup>2</sup> (Bson<sup>3</sup>). Utilizamos este sistema de bases de datos, ya que es flexible en el almacenamiento de documentos, tiene una sintaxis dinámica para hacer consultas y es de código abierto. Hasta el momento no encontramos proyectos específicamente artísticos que utilicen el sistema de base de datos MongoDB como parte de un proceso compositivo. Sin embargo, este tipo de bases de datos son muy utilizadas por distintas redes sociales, vale destacar a la plataforma SoundCloud<sup>4</sup> (2007), ya que es una red social exclusivamente de música y que utiliza MongoDB. Por nombrar otras

redes, LinkedIn, Facebook y Twitter utilizan este sistema.

Retomando el ejemplo de Soundcloud encontramos similitudes entre nuestro proyecto y esta red social. No solo en el tipo de base de datos que utiliza sino que existe también un vínculo en el que ambas están pensadas como una plataforma de audio abierta con la finalidad de contener obras y la interacción entre artistas. Otro punto en común con SoundCloud es la posibilidad de manipular los archivos que contiene esta plataforma, ya que posee una herramienta musical que se vincula con algunas DJ app, permitiendo utilizar todos los track del catálogo, *DJ integrations with SoundCloud*<sup>5</sup>. En esta línea existe una gran cantidad de aplicaciones pero nos centraremos en algunas. Encontramos proyectos como Bandcamp (2008) y Jamendo (2005) en los cuales personas han subido y compartido millones de archivos y tienen la posibilidad de realizar streaming en directo. Otro tipo de plataforma a la que, en alguno puntos, se acerca a nuestro trabajo es Freesound.org<sup>6</sup> (2005). La cual es un repositorio colaborativo de muestras de audio de todo tipo y una organización sin fines de lucro. MySpace<sup>7</sup> (2003) es una red social que permite crear un perfil y publicar cierta cantidad de pistas en este. Existe un proyecto en MySpace que nos parece bien relacionarlo con el nuestro. Este pertenece a Peter Traub<sup>8</sup> y se llama *ItSpace* (2007). En la primera versión de *ItSpace*, el artista altera la convención de la red social al crear nueve páginas de perfil pertenecientes a objetos cotidianos de su entorno. Cada página tenía una foto de perfil del objeto, una descripción y una pieza musical de un minuto compuesta por grabaciones del objeto golpeado y resonando de varias maneras. Los objetos eran amigos entre sí en dicha red, y se invitaba a los visitantes del sitio a crear nuevos perfiles del tipo *ItSpace* para hacerse amigos de los existentes. La segunda versión es una instalación interactiva del mismo año. Podemos ver similitudes de otro tipo en el proyecto *The Urban (Un) Seen—ai as Future Space*, de Bettina Zerza<sup>9</sup> y Tae Hong Park<sup>10</sup>, (2019, Bienal de Urbanismo y Arquitectura en Shenzhen, Hong Kong). Si bien es un proyecto centralmente urbanístico, es multidisciplinar. Este analiza el impacto que las actuales tecnologías digitales están teniendo en las ciudades y el desdibujamiento paulatino de la línea divisoria entre los espacios públicos. También abordan la urbanización sostenible e indagan en la problemática de la contaminación acústica urbana que acontece en las grandes ciudades. Gran parte

<sup>1</sup> NoSQL

<sup>2</sup> Json

<sup>3</sup> Bson

<sup>4</sup> SoundCloud

<sup>5</sup> *DJ integrations with SoundCloud*

<sup>6</sup> Freesound

<sup>7</sup> MySpace

<sup>8</sup> Peter Traub

<sup>9</sup> Bettina Zerza

<sup>10</sup> Tae Hong Park

de la información que utilizan está centrada en el ruido urbano. Atribuyen al ruido una perspectiva más amplia al añadir otras capas de información de las que normalmente se analizan: una capa sobre la movilidad urbana de personas, otra capa sonora en la que aparecen los patrones aurales de ese entorno, y una capa semántica / ar (realidad aumentada) / vr (realidad virtual) en donde hay información cualitativa de las emociones de estas personas al sentir esos espacios urbanos. El y la artista, dedicados a la arquitectura, la música y la programación, introducen dentro del espacio expositivo los rumores y flujos de la ciudad, visualizando, sonorizando, animando a la reflexión y a experimentar la ciudad. Utilizan sensores tanto en el interior como en el exterior de la exposición que recopilan datos previos y durante el evento. Las personas participan a través de sus móviles mediante una app con la que se mide el ruido de la ciudad y se experimentan las realidades aumentadas y virtuales de la pieza. El resultado de los datos y sus traducciones a imagen y sonido son las proyecciones audiovisuales de la obra. La red de sensores de ruido es *data-driven*, *community-driven* y *art-driven*, es decir, que los sonidos y las imágenes son dirigidos por el flujo de datos, las interacciones comunitarias y por la parte artística. La obra acontece dentro de la exposición como una instalación audiovisual. A esto se le agrega las propuestas de ar y vr experimentadas mediante los smartphones. El conjunto permite a las personas percibir la ciudad de una forma mixta y ampliada [9]. En concordancia proponemos analizar cómo los usuarios a través de sus experiencias cotidianas, pueden plasmar su entorno, sus emociones y sus acciones a través de la aplicación.

*Naked on Pluto* (2010-2015) es un proyecto de los artistas Dave Griffiths, Aymeric Mansoux y Marloes de Valk con el que vemos algunas aristas en común con el trabajo que estamos presentando. *Naked on Pluto* es un juego on line que está desarrollado con software libre, su jugabilidad se basa en una parodia de los rasgos invasivos de las redes sociales. La ciudad donde tiene lugar, *Elastic Versailles* es animada por la lógica combinatoria de una comunidad de 57 robots de inteligencia artificial que deducen datos de Facebook de los suscriptores del juego. Los robots *Naked on Pluto*, difícil de distinguir de los demás agentes en este entorno textual, está formada por guardianes disfuncionales. Los participantes del juego pueden romper los sistemas de control de acceso, pero los robots tienen una cierta flexibilidad para permanecer. Los jugadores pueden intentar anular las restricciones del juego y unirse para romper el sistema y escapar del mismo. Las actividades se describen en un blog y en Twitter, y los robots desbaratan la información del usuario y la de sus

contactos mientras incitan constantemente a clicar, declarar, señalar y comprar, generando más o menos enlaces espurios a una velocidad vertiginosa. Algo que conecta también con este proyecto es que en este juego ninguna información del jugador queda almacenada, ni se comparte ni se devuelve a Facebook en este ecosistema manejable en el que todo lo que cuenta son resplandores de una visibilidad fugaz. De esta manera, *Naked on Pluto* evidencia y denuncia la proliferación de agentes virtuales que recogen nuestros datos personales para reformular de modo insidioso nuestros entornos y perfiles virtuales, resaltando los contrastes ambivalentes de las redes sociales: los amigos como atractivos cuantificables y personas cuidadosamente a la moda que logran impartir un sentido de “intimidad”, y publicaciones nada ingenuas de datos “privados”.

Encontramos similitudes en la búsqueda de narrativas e intención estética en el manejo de los datos en varias obras de Nicholas Rougeux. De su larga lista de estas obras, tomaremos una en particular, ya que en ella toma piezas musicales y las convierte en imágenes abstractas. Sin embargo, se pueden ver patrones muy reconocibles y distintas conexiones a través de los esquemas que el artista logra. La obra en cuestión se llama *Making of Off the Staff* (2016). Podemos vincularla con nuestro trabajo ya que enviaremos datos desde un software de música para poder convertir un archivo a imagen, como veremos más adelante.

Un proyecto que vincula redes sociales, datos y Pure Data es uno de Ressidum<sup>11</sup> (Thomas Mayer) Donde realiza una demostración de sonorización de Twitter de una charla en Hackerbrücke Developer Make and Meet 2013. Este si bien no es musicalmente relevante, muestra una forma de sonorizar los datos, los tweets se procesan y sonorizar de la siguiente manera: el contenido se divide en valores ASCII, y cada valor crea un sonido de 1/10 s de duración; los caracteres no imprimibles hacen una pausa, todos los demás se asignan al valor MIDI respectivo (y luego se utilizan para controlar un oscilador sinusoidal; cuanto más largo sea el nombre de usuario, más fuerte será el sonido).

Por ultimo, traemos al proyecto *Lucarda.com.ar* (2009) de Lucas Cordiviola<sup>12</sup>, donde aloja su discografía solista. En ella existen varios discos producidos en Pure Data y con los que plantea escucharlos a través del software e interactuar con este y la programación de distintos patches aleatorios. Estas obras están acompañadas de argumentaciones musicales, instrucciones y de las

<sup>11</sup> Ressidum

<sup>12</sup> Lucarda

letras que compone para algunas piezas. De esta manera, se vincula con nuestro trabajo, ya que propone una composición, producción y escucha particular de la obra desde Pure Data. En esta página, entre otras cosas, también se encuentra un tutorial basto de Pure Data, el que es una traducción del original de Johannes Kreidler<sup>13</sup>.

Por lo expuesto, consideramos que nuestro proyecto podría aportar una visión de la problemática que involucra a la exposición en la que nos encontramos como personas usuarias de la red. En particular, este trabajo tendría que utilizar los mismos medios a los que de alguna manera denuncia. Por ejemplo, los datos de usuario. Esto hasta ahora nos es inevitable porque necesitamos de ellos para obtener resultados en una etapa de investigación. Podría ser posible así visibilizar las opciones que tienen las distintas entidades involucradas en acceder a un uso ético de las diferentes herramientas del medio tecnológico comunicacional.

## 2. RESUMEN DE DESARROLLO Y METODOLOGÍA.

### Resumen de desarrollo:

Para describir más claramente el proceso de la etapa inicial de este proyecto nos parece aclarador mencionar los objetivos planteados en el marco de BEFI (2020). En primer lugar propusimos recopilar y estudiar material bibliográfico, documentación de software y códigos de programación referentes a la temática del proyecto. Para luego desarrollar un script en Python<sup>14</sup> para extraer y almacenar datos en una base de datos NoSQL. Así como también, vincular Pure Data<sup>15</sup> (PD) y bases de datos NoSQL a través de librerías como Json, Socket<sup>16</sup>, Bson, Pickle<sup>17</sup>, utilizadas desde Python. Extraer y procesar datos en el entorno de programación PD; desarrollar un código para transformar en parámetros musicales los datos de personas usuarias obtenidos y convertirlos en archivo de audio; enviar la información a Processing a través del protocolo de comunicación OSC, desde PD. Desarrollar en Processing una serie de scripts para generar visuales a raíz de los datos recibidos desde PD. Almacenar archivos en documentos Json en una base de datos NoSQL desde PD y Processing, a través de Python. Desarrollar y diseñar una interfaz de usuario para la aplicación (GUI), en Python con

Django<sup>18</sup> o Kivy. Analizar musicalmente los archivos creados y compararlos con los datos enviados a Python. Documentar todo el proceso referente al proyecto. Realizar un documento del desarrollo y evolución de la red social. Poner la información y el proyecto en si a disposición del ámbito académico.

### Metodología:

Para alcanzar la etapa inicial de desarrollo del software y su API Rest, la que se basa en herramientas de código abierto, proponemos crear un script capaz de extraer y almacenar datos en bases de datos NoSQL de MongoDB, en Python mediante la librería llamada Pymongo. De este modo, podremos almacenar datos provenientes de eventos de cursor, emulando *touch* para probar su funcionamiento. Luego de poder almacenar y extraer datos, procederemos a realizar pruebas para que PD pueda recibir y enviar datos a las bases de datos, esto puede ser recibiendo documentos Json convertidos a bytes mediante librerías como Json, Bson, Sys y/o Pickle, enviados mediante la librería Socket, desde (y hacia) Python.

Una vez que PD reciba datos a través de las librerías mencionadas, podremos convertirlos a pequeños archivos de audio tanto en memoria ram, almacenados en arrays, como también en disco rígido, preferentemente en formato wav. Los guardaremos en tal formato a través de un algoritmo que crearemos en PD.

Si bien Processing recibirá y enviará datos desde PD utilizando Open Sound Control<sup>19</sup> (OSC), también exploraremos la posibilidad de trabajar en Processing en modo Python. No obstante, los datos provenientes desde Python hacia PD serán enviados a Processing, así como también, los valores resultantes de osciladores de Pure Data, y serán almacenados en variables para la modificación y movimiento de objetos de programación para generar visuales.

A su vez, en PD crearemos un patch para poder enviar los archivos de audio y otras peticiones hacia Python -también en paquetes de bytes- para que a través de la librería Pymongo genere documentos Json y los almacene en las bases de datos creadas para entregar a la persona usuaria un archivo de audio o uno visual.

Desarrollaremos una interfaz de usuario (GUI) básica para la aplicación móvil. Creemos conveniente utilizar Kivy debido a que es una librería de código abierto de Python, es una

---

<sup>13</sup> Johannes Kreidler.

<sup>14</sup> Python

<sup>15</sup> Pure Data

<sup>16</sup> Socket

<sup>17</sup> Pickle

---

<sup>18</sup> Django

<sup>19</sup> Open Sound Control (OSC)

herramienta capaz de desarrollar una interfaz multitáctil, puede acceder a los sensores de un dispositivo.

Planteamos una conexión cliente servidor entre el código de la app y el código del backend basado en una pc y conectado de forma remota a la base de datos Atlas. Nótese en Fig. 1 que proponemos que la app no realice - al menos algunas- peticiones al servidor para realizar algunas consultas, como por ejemplo visualizar sus datos almacenados o eliminarlos cuando desee. Si bien puede parecer contradictorio a la necesidad nuestra de recopilar datos para un análisis con respecto a la narrativa de datos. Pero consideramos tener en cuenta el concepto de este trabajo y que es el orden de prioridad que tenemos. Sin embargo, como se dirá más adelante, se pauta con un grupo de personas específicas para tal finalidad. Debemos también cuestionar y estudiar la seguridad de las personas y las bases de datos al permitir tales accesos, aunque sabemos que MongoDB Atlas posee una arquitectura que administra los datos en una instancia sin necesidad de un servidor.

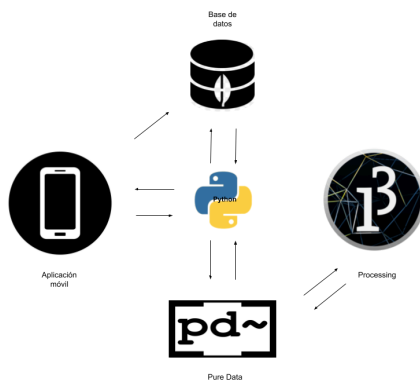


Figura 1. API Rest de etapa inicial.

En la segunda etapa comenzaremos un análisis de los archivos creados en relación al tren de datos que se recopilamos para tal fin. En primer lugar pautaremos con una determinada cantidad de personas para que interactúan cotidianamente con la aplicación para llevar a cabo una experiencia piloto. Entonces recopilaremos mediciones de distintos sensores y parámetros provistos por las personas usuarias mediante su dispositivo. Así podremos observar si es que los distintos tipos de comportamientos de las personas se reflejan su narrativa sonora y visualmente. También realizaremos un estudio comparativo entre las obras de usuaries y las existentes en el material bibliográfico referente con el fin de encontrar comportamientos generales y comportamientos

particulares o eventuales. Sobre los archivos realizaremos acciones de análisis como detectar patrones, trazar analogías, reconocer sucesos determinados e indeterminados en relación a los eventos. Desarrollaremos un espacio de comentario en las publicaciones para enriquecer la experiencia de usuarie y ganar datos para sumar al análisis propuesto.

Con respecto a la interfaz de usuario consideramos que debe tener login, logout, alta y baja de usuario. Interacción entre usuaries: comentar posteos, seguir a otros usuaries. Manipulación tanto sobre sus archivos como de los datos almacenados en las bases. Para esta interfaz se requerirá un backend dinámico por lo que se evaluará llegado el momento de desarrollo la incorporación de Django para la app, ya que es una framework potente para el desarrollo de aplicaciones y es de código abierto.

Realizaremos un documento con un análisis de toda la información obtenida acerca de la experiencia para publicar medios pertinentes. Esta investigación podría ser un aporte útil para otras áreas que exceden a este trabajo como psicología, medicina, salud, medioambiente, educación, ciencias sociales.

### 3. RESULTADOS

El resultado de la etapa inicial hasta el momento, comienza con la comunicación de Python con las bases de datos y PD, y de este último con Processing. Logramos guardar los datos en MongoDB en formato Json y de manera local, así como los de *login* y *touch* en Atlas de manera remota. Desarrollamos una interfaz de usuarie inicial (Figura 6, 7, 8) y otra posterior se encuentra en desarrollo mediante la librería MDScreem. Hicimos música con los datos recibidos en PD. Allí creamos algoritmos de composición musical para darle una estética sonora determinada a los datos recibidos, a través de efectos y técnicas de audio. Guardamos archivos en formato wav en la pc a través de PD. Estos archivos de audio pueden ser enviados a Python en paquetes de bytes para ser guardados en bases de datos. Desarrollamos un script para Processing.

Para la segunda etapa hasta ahora, pudimos realizar comparaciones rápidas entre el audio obtenido y el tren de datos que lo ha conformado y sacar conclusiones básicas acerca de la relación que existe entre sus narrativas. Podemos observar que los parámetros serán ajustables a los comportamientos de las personas. Por ejemplo: al deslizar el dedo por una pantalla táctil se genera un barrido de de datos que son consecutivos por lo cual podemos decir que es una sucesión



determinada de eventos. En cambio al utilizar una aplicación cualquiera la persona va posando su dedo de manera aparentemente aleatoria por lo que podemos referirnos a una sucesión indeterminada de eventos.

#### 4. DISCUSIÓN

Con respecto al desarrollo de la etapa inicial hemos alcanzado a desarrollar un script capaz de extraer y almacenar datos en una base de datos NoSQL mediante el sistema MongoDB y su librería que lo vincula a Python llamada Pymongo. De este modo, logramos almacenar datos provenientes de eventos de cursor, emulando *touch* e imagen. Luego de poder almacenar y extraer datos desde dichas bases de datos analizamos y realizamos pruebas para que PD pueda recibir y enviar datos a Python, recibiendo documentos Json convertidos a bytes, enviados mediante la librería Socket. También utilizamos y exploramos otras librerías para la conversión de los paquetes de datos como, Pickle, Bson y Sys<sup>20</sup>. La utilización de estas ha sido determinante para la conexión de los distintos lenguajes, transmisión de datos, conversión de los formatos de los mismos y la ejecución sonora. La librería Sys no es nombrada en la etapa inicial pero fue necesaria su inclusión posterior y utilización junto a Socket para crear un canal de recepción de datos en Python, enviados desde PD. (Fig. 2)

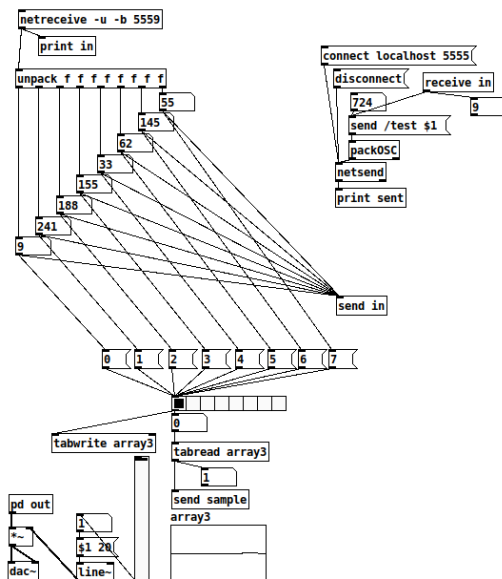


Figura 2. A la izquierda Pure Data recibe datos desde Python (netreceive), a la derecha los envía (netsend).

Una vez que logramos que PD reciba datos a través de las librerías mencionadas, fue posible convertirlos a archivos de audio tanto en memoria ram, almacenados en arrays, como también en el disco rígido, en formato wav. Los últimos fueron guardados en tal formato a través de un algoritmo creado en PD (Fig. 3).

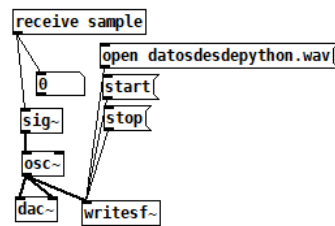


Figura 3. Algoritmo que guarda un archivo de audio en disco rígido.

Para la comunicación y transmisión de datos entre Pure Data y Processing utilizamos Open Sound Control<sup>21</sup> (OSC). OSC es un protocolo de comunicación para dispositivos y aplicaciones que optimiza la transmisión de datos digitales a través de una red. En este caso la red también es local, ya que comunicamos por un puerto específico las aplicaciones PD y Processing mediante la librería OSC. Elegimos este protocolo porque es efectivo, hay librerías y documentación suficiente. De esta manera, fue posible establecer una comunicación exitosa, los datos provenientes desde Python hacia PD fueron enviados a Processing, así como también, los valores resultantes de osciladores de Pure Data, y fueron almacenados en variables para la modificación y movimiento de objetos de programación para generar visuales en tiempo real (Fig. 4).

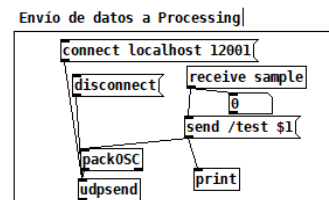


Figura 4. A través del objeto *udpsend* se envían paquetes de datos hacia Processing, creados por *packOSC*.

A su vez, desde PD debíamos poder también enviar datos hacia Python para que a través

<sup>20</sup> Sys

<sup>21</sup> [Open Sound Control \(OSC\)](#)

de la librería Pymongo cree colecciones de datos en documentos Json y los almacene en las bases de datos remotas para entregar a la persona usuaria un archivo de audio o un archivo visual generado con sus datos ingresados a la base de datos. En Fig. 5 se puede observar un recorte de ambos algoritmos, donde Python recibe datos desde PD:

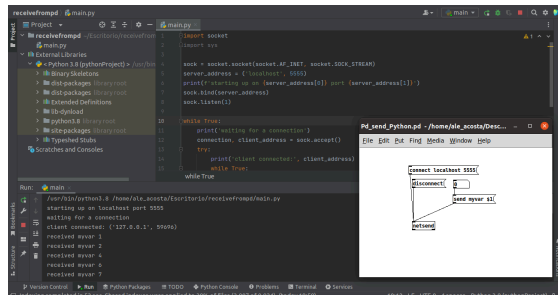


Figura 5.

Llegados a este punto, tanto Python como PD envían y reciben datos; que este último los envía a Processing; y que las bases de datos de MongoDB también envían y reciben datos. En esta instancia de desarrollo comenzamos a crear una interfaz de usuario (GUI) básica para una aplicación, optamos por utilizar Kivy debido a que es una librería de código abierto de Python, es una herramienta capaz de desarrollar una interfaz multitáctil, puede acceder a los sensores de un dispositivo, permite un rápido desarrollo frontend y está respaldada por una amplia documentación y una comunidad activa. Con respecto a la instancia de desarrollo de la GUI, la interfaz hasta esta instancia fue realizada toda en Kivy.

La aplicación inicia en una página inicial de *login* (Fig. 6); luego tiene otra donde el usuario accede a opciones (acciones) a través de botones, como activar el envío y almacenamiento de datos a una base de dato “Dibujar un sonido”, ver dichos datos “Mostrar datos de dibujo”, eliminarlos “Remover datos de dibujo”, salir de la aplicación “Log out” (Fig. 7) y una ventana emergente (pop) “touch\_sound” donde se presenta un área para el ingreso de los datos *touch* o cursor (Fig. 8).

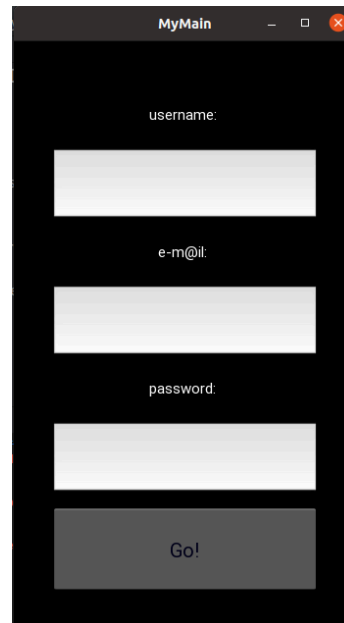


Figura 6.

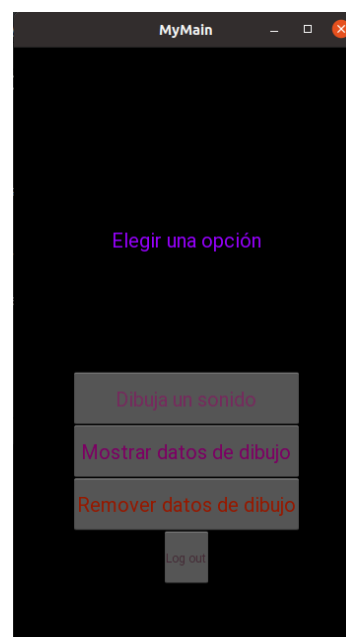


Figura 7.

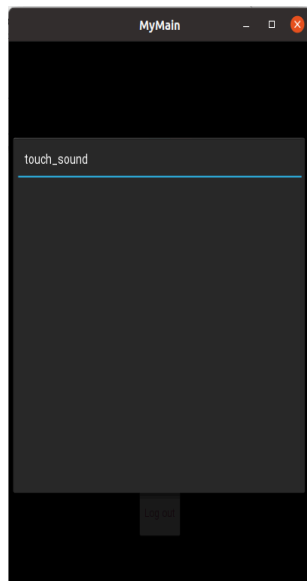


Figura 8.

Finalmente se tuvo en cuenta la utilización de Django para la parte de servicios y foma parte del backend debido a su dinamica, a que es posible su trabajo en Python y es open source. La librería Kivy implementada fue suficiente para la instancia de frontend para este proyecto. Sin embargo, más adelante esta interfaz la hemos reemplazado por otra librería perteneciente a KivyMD, llamada MDScreen.

Los datos de usuarie generados a través de la aplicación y los datos generados mediante el trabajo de Python (junto con Pure Data y Processing), son almacenados en la base de datos remota en un cluster gratuito (aunque limitado a 500 Mb) de MongoDB llamada Atlas. En la figura 9 se puede ver esta base de datos donde los nombres refieren directamente a los de las variables de bases de datos y colecciones declaradas en el script de Python (llamadas login, users, dbtouch, coltouch).

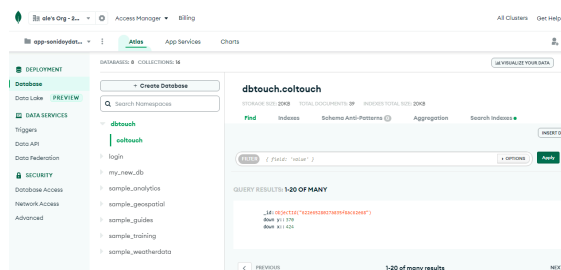


Figura 9.

Con respecto al *backend*, se realizó en Python, siendo este un centralizador apto de las operaciones y peticiones requeridas tanto por el *frontend*, las aplicaciones externas como PD y Processing, así

como por las articulaciones con las bases de datos tanto locales como remotas a modo de API Rest.

Para la compilación en un archivo de lo descrito hasta el momento usamos las librerías Cython<sup>22</sup>, Java Script<sup>23</sup>, Android, NDK<sup>24</sup>, SDK<sup>25</sup>, Python for Android<sup>26</sup>, Pymongo, Kivy y para la construcción de un archivo .spec<sup>27</sup> (inicio del .apk) se utiliza Buildozer, esta última herramienta requiere de todas las librerías y software recién mencionados, en sus versiones correspondientes para alcanzar la construcción del archivo .spec. Este archivo .spec se debe configurar, especificando requisitos para la compilación de un archivo .apk<sup>28</sup> que luego se instala en un dispositivo android para ejecutar la aplicación a través de un comando en la terminal, en este caso de Ubuntu 20.04.

## 5. CONCLUSIONES

El resultado que obtuvimos de la etapa inicial hasta el momento, comenzó con la comunicación entre Python, las bases de datos y PD, y de este último con Processing. Logramos guardar los datos en MongoDB en formato de documentos Json y de manera local, así como los de *login* y *touch* en Atlas de manera remota. Desarrollamos una interfaz de usuarie inicial y otra posterior se encuentra en desarrollo mediante la librería MDScreen de KivyMD. Hicimos música con los datos recibidos en PD. Allí creamos algoritmos de composición musical para darle una estética sonora subjetiva a los datos recibidos, a través de técnicas de audio y efectos. Guardamos archivos en formato wav en disco rígido a través de PD (Fig. 10). Estos archivos de audio pudieron ser enviados a Python en paquetes de bytes para ser guardados en bases de datos. Desarrollamos un script para visuales en Processing.

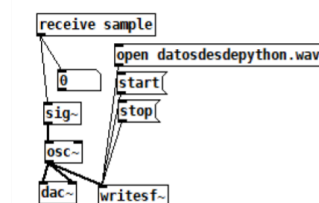


Figura 10.

<sup>22</sup> [Cython](#)

<sup>23</sup> [Javascript](#)

<sup>24</sup> [NDK](#)

<sup>25</sup> [SDK](#)

<sup>26</sup> [Python-for-android](#)

<sup>27</sup> [Archivo .spec](#)

<sup>28</sup> [Archivo .apk](#)



Para la segunda etapa hasta ahora, pudimos realizar comparaciones rápidas entre el audio obtenido y el tren de datos que lo ha conformado y sacar conclusiones básicas acerca de la relación que existe entre sus narrativas. Podemos observar que los parámetros serán ajustables a los comportamientos de las personas. Por ejemplo: al deslizar el dedo por una pantalla táctil se genera un barrido de datos que son consecutivos por lo cual podemos decir que es una sucesión determinada de eventos. En cambio, otro ejemplo sería al utilizar una aplicación cualquiera la persona va posando su dedo de manera aparentemente aleatoria por lo que podemos referirnos a una sucesión indeterminada de eventos. Si una persona permite recopilar datos en segundo plano de estos eventos touch, podremos obtener distintos tipos de sucesiones de datos, que variarán según su comportamiento.

Todas las pruebas de funcionalidad del proyecto y que se informaron aquí, las hemos realizado en una laptop i3, so ubuntu 20.04 en la cual se pudo probar la correcta funcionalidad estructural de este desarrollo. Las bases de datos por su parte funcionaron tanto en modo *localhost* como remoto a través de Atlas.

Hemos llegado a generar el archivo .apk al ser compilado por Buildozer y fue instalado en un dispositivo Android API 30. Hasta el momento en el dispositivo móvil, la app abre en el dispositivo mostrando su página de inicio, la cual contiene y permite el llenado de campos de la instancia de login (username, email, contraseña y un botón de acceso). Sin embargo, es posible corroborar su funcionamiento desde la computadora donde fue desarrollada.

Como una tentativa inicial creamos en Django una plataforma para red social donde es posible registrarse, autenticar, crear un perfil, interactuar entre usuarios y subir información generada. Optamos por utilizar este framework para estos servicios y también proyectando a este trabajo a tener en algún momento una versión web. (Fig. 11, 12, 13).

Actualmente continuamos trabajando en el desarrollo de la app así como tenemos presente las siguientes ampliaciones del proyecto: reforzar la seguridad en la autenticación de la persona usuaria; una interfaz más intuitiva y dinámica mediante la librería MDCard; recopilación de datos en segundo plano con permiso de parte de las personas usuarias y acceso y manipulación de los datos almacenados; utilización de la librería PurestJson<sup>29</sup> para enviar documentos a puredata sin ser serializados;

investigar la librería pd-for-android<sup>30</sup>; acceso a distintos sensores a través de Kivy y las API de los dispositivos; modificación de parámetros musicales y visuales a través de la app; procesamientos en tiempo real para proyecciones performáticas; estudio de la seguridad a través de peticiones sin servidor a Atlas; un sitio web donde se exponga todo el material perteneciente al proyecto; en relación a la problemática planteada, vemos posible dar nuestra visión una temática que atraviesa a todas las personas.

Sabemos que cada vez que navegamos en la web, utilizamos o instalamos una aplicación en un dispositivo en la mayoría de los casos, cedemos datos personales. Es creciente la cantidad de detalles de nuestras vidas que dejan un rastro digital en constante actualización. En tanto que almacenar, gestionar y utilizar datos, en algunos casos con fines lucrativos, plantea renovados problemas relacionados con la privacidad, la propiedad, la identidad, la intimidad, la confianza o la reputación. Es por esto que encontramos un campo propicio para la exploración artística y, mediante una aplicación con posibilidades de una constante expansión, buscamos visibilizar esta problemática actual. Por el otro, pudimos evidenciar que hay una analogía latente entre la vida cotidiana y el arte de modo que los datos son un aporte directo como material compositiva para cualquier arte que decida incorporarlo.

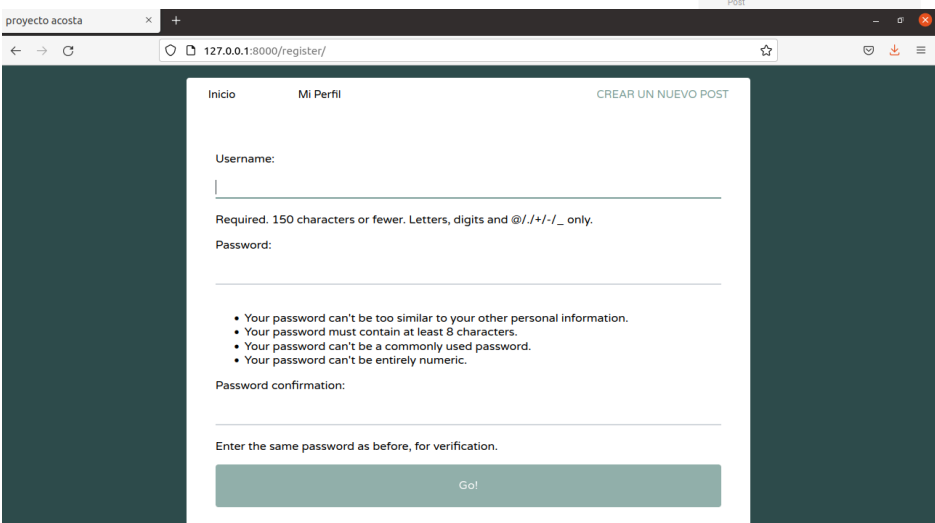
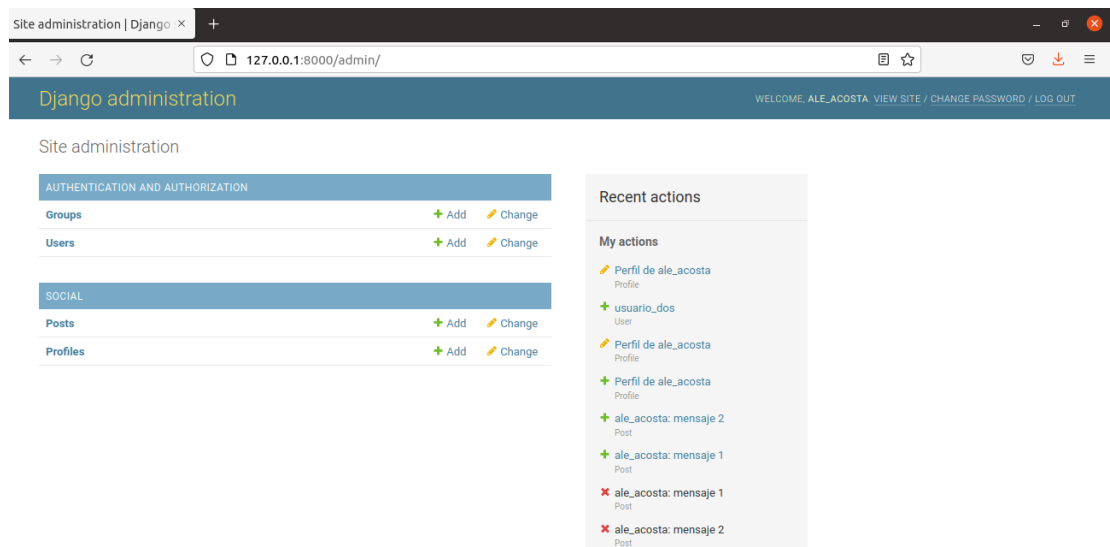
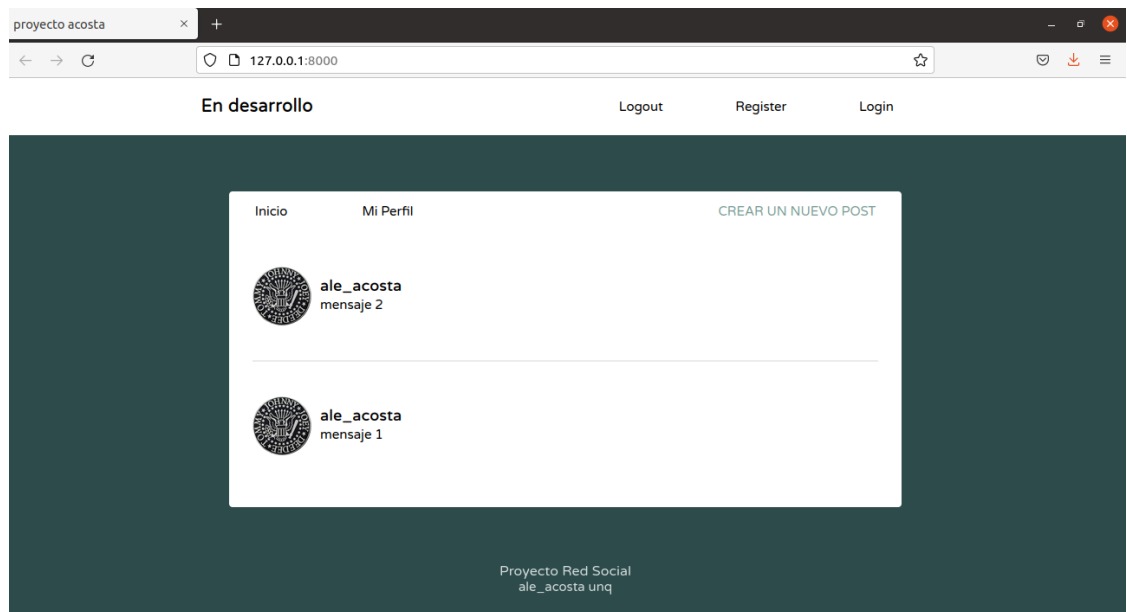
Por último, el análisis en este trabajo podría ser de utilidad, como un aporte para otras áreas de investigación que exceden a este trabajo como son psicología, medicina, salud, medioambiente, educación, ciencias sociales, entre otras.

---

<sup>29</sup> PurestJson

---

<sup>30</sup> pd-for-android



Figuras 11, 12, 13.

## 6.REFERENCIA

- [1]Morgado, B., & López, E. (2016). Arte, sonido e internet: el net. sound art frente a la experimentación sonora de las vanguardias y neovanguardias. *Universum (Talca)*, 31(1), 191-208.
- [2]Weaver, S. D., & Gahegan, M. (2007). Constructing, visualizing, and analyzing a digital footprint. *Geographical Review*, 97(3), 324-350.
- [3]Colmenarejo Fernández, R. (2017). Una ética para Big data: introducción a la gestión ética de datos masivos. *Una ética para big data*, 1-130.
- [4]Boyd D., Crawford K. (2012). Critical Questions for Big Data. (Pag. 662-679).
- [5]Sibilia, P. (2017). ¿ Qué es privado en la era de la extimidad?: de la confesión intimista a la curaduría de sí mismo. *Docta: Revista de Psicoanálisis*, 134-149.
- [6]Lupi, G. (2015). The New Aesthetic of Data Narrative. In: Bihanic, D. (eds) New Challenges for Data Design. Springer, London. [https://doi.org/10.1007/978-1-4471-6596-5\\_3](https://doi.org/10.1007/978-1-4471-6596-5_3)
- [7]Ciuccarelli, P., Lupi, G., & Simeone, L. (2014). Visualizing the data city: social media as a source of knowledge for urban planning and management. Springer Science & Business Media.
- [8]Lupi, G., & Posavec, S. (2016). Dear data. Chronicle books.

[9]Rougeux, N. (2020). Off the Staff: An Experiment in Visualizing Notes from Music Scores. *Patterns*, 1(3), 100055.

[10]Nicholas Rougeux. Works of Nicholas Rougeux. <https://www.c82.net/> [2022]

[11]Traub, P. M. (2011). ItSpace. In ACM SIGGRAPH 2011 Art Gallery (pp. 376-377).

[12] Stefanie Posavec. Writing Without Words <http://www.stefanieposavec.com/writing-without-words> [2022]

[13] astrovandalistas. La Sonora Telemática Transito\_MX 06, México City (September 2015) <https://astrovandalistas.cc/ast/>  
[2022]

