UNIVERSIDAD CATÓLICA DE LA SANTÍSIMA

CONCEPCIÓN

FACULTAD DE INGENIERÍA

DESARROLLO DE PLATAFORMAS DE INTELIGENCIA DE NEGOCIOS Y PROCESAMIENTO DE DATOS EN LA EMPRESA DANA SPA

para optar al título de Ingeniero Civil Industrial

SAMUEL JESUS OSORIO CARRIMAN

DANA SPA

Nombre Supervisor Empresa	Nombre Profesor Evaluador 1	Nombre Profesor Evaluador 2
Carlos Navarrete	Claudia Carrasco	Gonzalo Bordagaray

Nota Informe escrito:

Dedicatoria

A mis padres Mauro y Roxana, a mi hermana Catalina, y a mis amigos, quienes estuvieron conmigo desde el principio y quienes aparecieron en el camino.

Resumen ejecutivo

El presente informe tiene por objetivo explicar la realización de una plataforma web, más específicamente el Backend de dicha plataforma, entregar guía de los pasos para su desarrollo y las tecnologías utilizadas en él de manera que puedan ser replicadas en futuros proyectos. En la actualidad se está rodeado de una gran cantidad de datos que no están siendo utilizados y de los cuales es posible obtener, mediante análisis estadísticos y modelos matemáticos, información para una toma de decisiones basada en evidencia, para ello la Ciencia de Datos (Data Science) proporciona herramientas que facilitan el proceso de obtención de datos, su procesamiento y posterior análisis para generar conocimiento. Para demostrar el impacto que tiene Data Science, ya sea en la academia o en la industria, se exponen tres proyectos en los cuales se aplica, CIS2, plataforma de difusión de innovación que es realizada por las empresas del país, Open Tech Biobío, evento para generar vínculos entre el ecosistema de innovación de la Región del Biobío y empresas que buscan soluciones a sus problemas, y HOSPITAL GO, proyecto ganador del Desafío de Salud convocado por la Aceleradora de Negocios de la Universidad del Desarrollo, UDD Ventures. En cada proyecto se revisarán las actividades realizadas de las cuales se fue responsable, los conocimientos adquiridos y tecnologías aplicadas, además se provee el código fuente que hace posible la ejecución de la plataforma y enlaces para acceder a las mismas en todos los proyectos mencionados.

Evaluación intermedia

Rúbrica de Evaluación Intermedia de Práctica Profesional Tutelada

Datos de la Empresa/Organización

Nombre o Razón Social	DANA SPA
Dirección	BALMACEDA 316, CONCEPCION
Giro	PROCESAMIENTO Y ALMACENAMIENTO DE DATOS
Nombre Supervisor de Práctica	CARLOS CAMILO NAVARRETE LIZAMA
Cargo	LIDER EN BIGDATA E INTELIGENCIA DE NEGOCIOS
Profesión	INGENIERO CIVIL INDUSTRIAL
Fecha de la Evaluación	18 DE OCTUBRE DE 2018

Datos del Estudiante

Nombre Completo	SAMUEL JESUS OSORIO CARRIMAN
Rut	18.876.905-5
Teléfono de contacto	+56956210444
Correo electrónico	SOSORIO@ING.UCSC.CL

La Carrera cree firmemente en el trabajo conjunto con el medio externo, para formar profesionales actualizados en la disciplina, capaces de destacar por sus sólidos conocimientos, habilidades interpersonales y por el sello identitario otorgado la UCSC.

La información entregada por usted es utilizada como retroalimentación para validar y/o actualizar el perfil de egreso, así como para mejorar el quehacer académico de nuestra Carrera.

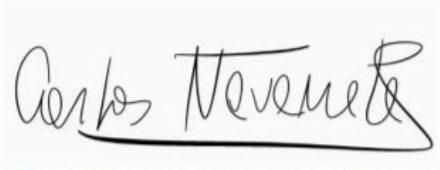
Marque con una X la calificación correspondiente a cada ítem
Si, debido a las actividades asignadas al estudiante, alguno de estos aspectos no pudo ser observado durante el periodo de práctica profesional tutelada, favor evaluar dicho ítem como "No observado".

	Excelente	Muy Bueno	Bueno	Regular	Deficiente	No Observado
Aporte Personal y Motivación	X					
Asistencia y puntualidad	X					
Responsabilidad	X					
Adaptabilidad	X					
Iniciativa	X					
Actitud para trabajar en equipo	X					
Relaciones Humanas	X					
Dominio de temas técnicos	X					
Capacidad para concebir soluciones	X					
Capacidad de respuesta ante requerimientos específicos.	X					
Capacidad de análisis y sentido común a la hora de resolver un problema.	X					
Claridad en la exposición de sus opiniones, ideas y argumentos	X					

Por favor incluya brevemente comentarios que permita mejorar el desempeño del estudiante en el periodo que resta de la práctica.

Samuel es un profesional proactivo, que posee una rápida capacidad de adaptación a nuevas tecnologías. En este periodo ha demostrado excelentes *skills* en torno a trabajar en proyectos ligados a la programación que requieren amplias habilidades de comprensión sistémica.

En lo que resta de práctica proyectamos que aplique conocimientos de análisis estadístico multivariado en modelos de Machine Learning y/o regresiones.



CARLOS CAMILO NAVARRETE LIZAMA
18.231.617-2

Nombre y Firma del Supervisor
Timbre de la Empresa

Esta pauta debe ser completada y firmada por el supervisor directo del alumno en práctica y entregada al Coordinador de Prácticas de la Carrera.

Evaluación final

Rúbrica de Evaluación Final de Práctica Profesional Tutelada

Datos de la Empresa/Organización

Nombre o Razón Social	**DANA SPA**
Dirección	**BALMACEDA 316, CONCEPCIÓN**
Giro	**PROCESAMIENTO Y ANÁLISIS DE DATOS**
Nombre Supervisor de Práctica	**CARLOS NAVARRETE LIZAMA**
Cargo	**LÍDER EN BIGDATA E INTELIGENCIA DE NEGOCIOS**
Profesión	**INGENIERO CIVIL INDUSTRIAL**
Fecha de la Evaluación	**07-01-2018**

Datos del Estudiante

Nombre Completo	**SAMUEL JESÚS OSORIO CARRIMAN**
Rut	**18.876.905-5**
Teléfono de contacto	**+569 5621 0444**
Correo electrónico	**sosorio@ing.ucsc.cl**

La Carrera cree firmemente en el trabajo conjunto con el medio externo, para formar profesionales actualizados en la disciplina, capaces de destacar por sus sólidos conocimientos, habilidades interpersonales y por el sello identitario otorgado la UCSC.

La información entregada por usted es utilizada como retroalimentación para validar y/o actualizar el perfil de egreso, así como para mejorar el quehacer académico de nuestra Carrera.

Marque con una X la calificación correspondiente a cada ítem
Si, debido a las actividades asignadas al estudiante, alguno de estos aspectos no pudo ser observado durante el periodo de práctica profesional tutelada, favor evaluar dicho ítem como "No observado".

	Excelente	Muy Bueno	Bueno	Regular	Deficiente	No Observado
Aporte Personal y Motivación	X					
Asistencia y puntualidad	X					
Responsabilidad	X					
Adaptabilidad	X					
Iniciativa	X					
Actitud para trabajar en equipo	X					
Relaciones Humanas	X					
Dominio de temas técnicos	X					
Capacidad para concebir soluciones	X					
Capacidad de diseñar	X					
Capacidad para implementar	X					
Capacidad para operar sistemas	X					

Capacidad de respuesta ante trabajo bajo presión	X					
Capacidad de cumplir satisfactoriamente, en términos de plazo y calidad, con los trabajos asignados.	X					
Capacidad de análisis y sentido común a la hora de resolver un problema.	X					
Claridad en la exposición de sus opiniones, ideas y argumentos	X					

Las preguntas siguientes no tienen puntaje asignado y por ende no influyen en la calificación final de alumno; sin embargo, para la carrera de Ingeniería Civil Industrial es importante que el supervisor las contesten honestamente con el fin de conocer si nuestros alumnos están respondiendo a las necesidades de la empresa; y en el caso contrario poder tomar las medidas correctivas para que ello ocurra.

Si le hiciera falta personal, ¿contrataría al estudiante que ha tenido en práctica?	**SÍ (X)**	**Indique el porqué** Como *StartUp* hemos contratado a Samuel, dado que se adapta de buena manera al trabajo de equipo, es responsable y metódico.
	NO	
¿Volvería a tener un estudiante en práctica de la UCSC?	**SÍ (X)**	**Indique el porqué** En términos generales, nos quedamos con buena percepción de la formación de profesionales que hacen dentro de la UCSC, por lo que en un futuro veríamos como un aporte un estudiante en práctica de la misma casa de estudios.
	NO	

Por favor incluya brevemente comentarios generales (positivos y/o negativos) sobre las actividades realizadas, el desempeño en el trabajo y su apreciación personal respecto al alumno en práctica profesional tutelada.

Samuel desempeñó sus labores dentro de la empresa con distinción máxima. Siempre buscó ser un aporte a los proyectos de la organización, impactando de manera destacada en el área de Ciencia de Datos. Si bien es de personalidad introvertida, estoy convencido que es un excelente profesional, que siempre cumple con las responsabilidades que adquiere. Dentro del proyecto HospitalGO por ejemplo, generó el inicio de un cambio para los pacientes de la comuna de Chiguayante, debido a las herramientas que construyó para entender de mejor manera a quienes asistían por atención de urgencia al SAR Chiguayante y SAPU Leonera.

Carlos Navarrete Lizama

Nombre y Firma del Supervisor
Timbre de la Empresa

Esta pauta debe ser completada y firmada por el supervisor directo del alumno en práctica y entregada al Coordinador de Prácticas de la Carrera.

Cc: Coordinador de Prácticas
Interesado
Archivo

Índice de contenido

Índice de figuras

Capítulo 1: Introducción

Los datos siempre han estado presentes en nuestra sociedad, cada vez que realizamos una compra, una búsqueda en internet, leemos un artículo, y hasta la cantidad de páginas que tiene este informe son datos, estos se encuentran en todas partes y de distintas formas, pero solo se quedan allí, almacenados sin un fin determinado, la transformación digital ha cambiado este panorama convirtiendo a los datos en la materia prima de muchos proyectos del denominado Internet of Things (IoT), lo que ha sido posible gracias a los avances tecnológicos que soportan el almacenamiento de Big Data y las redes capaces de transmitirlos en tiempo real. El cambio de este paradigma se debe a la valorización de los datos, los cuales, de ser transformados, se convierten en información que permite la posterior generación de conocimiento, de esta manera se desarrollan plataformas web que automatizan la extracción, transformación y carga de datos de interés para el usuario, logrando un panel que visualiza información oportuna para tomar decisiones basadas en evidencia. El proceso previamente expuesto es lo que se conoce como Data Science (Ciencia de Datos), cuyo objetivo es generar conocimiento de los datos con los que se cuenta, un área particular de esta ciencia es Business Intelligence (Inteligencia de Negocios), la cual está orientada a optimizar la toma de decisiones en los negocios, generando un panel de control que visualiza los descubrimientos obtenidos de los datos.

Objetivo general

El objetivo general del estudio es desarrollar plataformas de Inteligencia de Negocios y procesamiento de datos para la empresa DANA SpA. Para lograr dicho objetivo se plantean los siguientes objetivos específicos.

- Transformar los datos provistos al formato Tidy Data.
- Diseñar base de datos relacional, esquemas que contienen las entidades, atributos y relaciones, que representan la estructura de las tablas de datos y su relación con otras tablas que complementan su información.
- Diseñar Backend de plataformas de Inteligencia de Negocios, una capa lógicá que permite al Frontend ejecutar aplicaciones (API) diseñadas en el Backend.
- Implementar prototipos de plataformas de Inteligencia de Negocios.

Metodología de trabajo

En base a los objetivos expuestos se investiga la metodología para desarrollar una plataforma web y se selecciona el lenguaje de programación Python, el cual cuenta con una curva de aprendizaje pronunciada por lo tanto es relativamente fácil de entender y aplicar; Python es seleccionado puesto que existen librerías (conjuntos de funciones orientados a un proceso determinado) diseñadas para gestionar y producir un proyecto de plataforma web, además de administrar y gestionar la base de datos, y el servidor que mantendrá disponible dicha plataforma a los usuarios. Python es igualmente utilizado para aplicar Data Science y cuenta con librerías adicionales que automatizan las tareas para generar conocimiento, un ejemplo de esto es Prophet desarrollado por Facebook que contiene funciones para realizar predicciones en series de tiempo, Pandas una librería para limpiar y transformar los datos con los que se cuenta y los frameworks Django y Flask que están orientados a desarrollar el Backend de un proyecto de plataforma web. Data Science es empleada para realizar las actividades de Extraer, Transformar y Cargar los datos, además de encontrar tendencias y predecir haciendo uso de algoritmos de Machine Learning. La metodología mencionada es utilizada por igual en los tres proyectos que se exponen en este trabajo, especificando que aspectos son empleados para cada proyecto en particular.

Antecedentes generales de la empresa

La empresa donde se desarrolla la práctica tutelada es DANA SpA y su giro es Procesamiento de datos y actividades relacionadas con bases de datos, DANA SpA es una empresa de tecnología ganadora del Desafío Salud de la Aceleradora de Negocios de la Universidad del Desarrollo, UDD Ventures, para apoyar emprendimientos de la Región del Biobío enfocados en el área de la salud, que resuelvan problemas reales que se hayan detectado en las empresas e instituciones de salud participantes, teniendo en cuenta un modelo de negocios escalable con impacto en la Región, esto con el proyecto HOSPITAL GO cuyo objetivo es facilitar la utilización de datos para la toma de decisiones basada en evidencia en los centros de salud.

La misión de DANA es:

"*Apoyar el proceso de transformación digital mediante la entrega de soluciones al manejo de grandes volúmenes de datos, desarrollando plataformas web y tecnología de vanguardia que apunten a simplificar, ordenar y graficar la información para apoyar la toma de decisiones estratégicas en todo tipo de organizaciones.*

Nuestros servicios se adaptan a las necesidades de cada proyecto, proporcionando un manejo rápido de herramientas digitales y constante vinculación con el cliente, lo que permite asesorar su idea y guiarla en su óptimo desarrollo hasta obtener un producto moderno, seguro y eficiente."

La visión de DANA es:

“Consolidarse como una empresa tecnológica reconocida por su aporte en la transformación digital de empresas nacionales y extranjeras, destacando por la excelencia del trabajo desarrollado y por acercar nuevas tecnologías de manejo de información a las personas. “

La unidad en la cual se desempeñan las funciones es la de Inteligencia de Negocios, como se trata de una Startup que se encuentra en la fase de crecimiento, la empresa necesitaba capacitar recursos humanos en las nuevas tecnológicas, formando un equipo multidisciplinario que pueda afrontar proyectos simultáneos de variadas áreas, dentro de ellas salud, a esta unidad se aportó con conocimientos de optimización y toma de decisiones, generando herramientas que permitan procesar y facilitar la visualización de datos con los que se cuenta, obteniendo información y en última instancia logrando conocimiento con el uso de técnicas de predicción y generación de clúster en base a algoritmos de Machine Learning.

Capítulo 2: Antecedentes Generales

La revisión de la literatura inicia con investigaciones orientadas al proceso general para disponer de los datos, Extraction, Transformation and Loading Data (Gajare, P. Rangdale, S. 2015) presenta una metodología para, traduciendo el título del artículo, **Extraer**, obtener datos de distintas fuentes, puesto que pueden encontrarse en diversidad de archivos cuando se almacenan localmente, o en una base de datos, en cuyo caso se debe contar con credenciales para obtener los datos remotamente; **Transformar**, realizar operaciones en los datos que permitan homogeneizar su contenido, tratar casos especiales como datos atípicos e incluso la ausencia y formato de los mismos, se debe considerar tanto la estructura como el contenido de los datos; **Cargar**, una vez estén los datos estén preparados, estos son enviados a una base de datos para ser almacenados, y disponer de ellos cuando se les requiera. ETL expone el proceso para disponer de los datos desde un punto de vista general, pero las etapas son, por si mismas, casos que deben ser analizados particularmente. Realizar la transformación de los datos implica dedicar una gran cantidad de tiempo a analizar el formato de ellos y su estructura, lo que no es una tarea menor considerando que más del 80% del tiempo de trabajo se emplea en limpiar los datos (Kandel et al., 2011), en esta etapa existe un trabajo denominado **Data Wrangling**, que expone una colección de los principales desafíos que aparecen en los proyectos orientados a los datos, dentro de él se encuentran aspectos de documentación del trabajo que se realiza cuando se limpian los datos, tolerancia a los errores de los datos, aspectos de la calidad de los datos, el “ruido” que pueden contener, y la edición de los mismos. Al referirse al “ruido” que pueden

contener los datos se considera que estos pueden estar incompletos y/o erróneos, por ejemplo, si se considera una tabla que contiene datos de personas, como su edad y dentro de uno de los registros se encuentra una letra “X”, esto puede parecer trivial, pero en la práctica implica que la columna tenga un formato de texto puesto que el ordenador intentara darle un formato homogéneo a todos los datos, y puesto que no puede convertir la letra “X” en un número, la columna adquiere el tipo de formato texto, por ende, no se podrían realizar operaciones matemáticas como obtener el promedio de las edades, así también se pueden encontrar discrepancias y datos inconsistentes en datos de tipo texto, como nombres o códigos de regiones, en general, los datos serán tan diversos como las personas quienes los crean, lo que se ve afectado por su cultura, su país de residencia e incluso su educación.

Realizar **Data Wrangling** implica diseñar un algoritmo que tome los datos que son la materia prima del trabajo y los procese de tal forma que estén listos para ser utilizados con el formato y forma requerida, por lo tanto, se emplea la metodología que transforma la estructura de los datos, **Tidy Data**, que propone principios para estructurar los datos antes de ser cargados en algún sistema de almacenamiento, lo que logra facilitar la manipulación, modelado y la posterior visualización de datos. La estructura de **Tidy Data** tiene relación con el almacenamiento de las bases de datos relacionales, en las cuales cada columna es una variable y cada fila representa una observación, de esta forma se optimizan operaciones básicas como ordenar datos, filtrar, además de realizar transformaciones posteriores y obtener indicadores de utilidad.

Los principios que plantea son los siguiente:

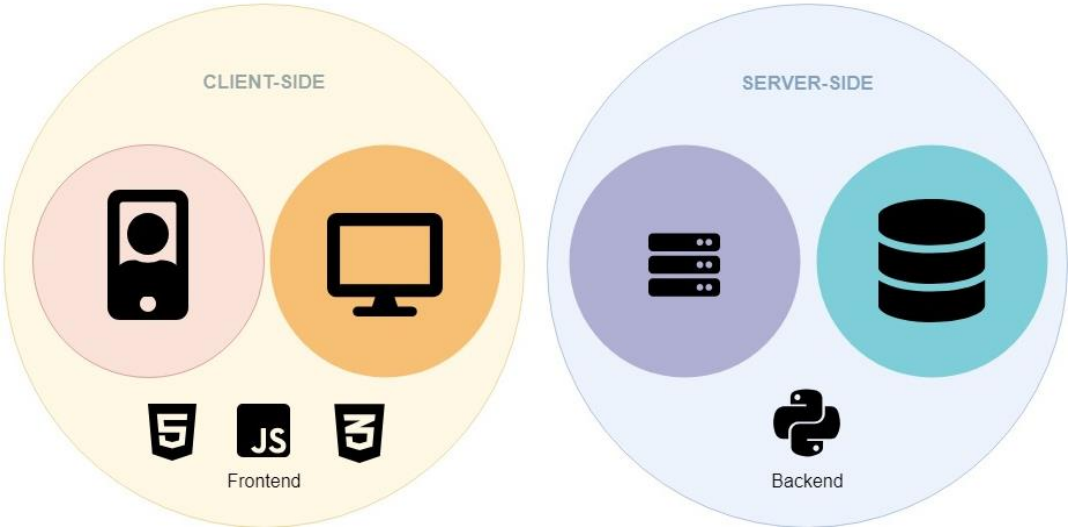
1. Cada variable forma una columna.
2. Cada observación forma una fila.
3. Cada tipo de unidad observacional forma una tabla.

Fuente: Wickham, H. (2014).

En posesión de los datos con el formato y estructura adecuados, estos deben estar disponibles para ser utilizados remotamente por la aplicación web, para disponer de los datos se necesita cumplir con dos aspectos fundamentales, la base de datos, que almacenará los datos, y el servidor, que actuará como intermediario entre el usuario y la base de datos, recibiendo las solicitudes de datos; una solicitud es equivalente a ingresar una dirección en el navegador web (Chrome, Mozilla Firefox, Safari, entre los más utilizados) como https://www.google.cl, solo que estos enlaces no están visibles explícitamente para el usuario. La unión de la administración de la base de datos y la gestión de las solicitudes hacia el servidor se conoce como Backend, y contempla todo el software necesario para acceder a los datos. Respecto del Backend, la metodología para estructurar el proyecto se basa en dos frameworks, Django y Flask; un framework es un conjunto de funciones diseñadas y estructuradas para lograr un objetivo particular, Django y Flask son frameworks orientados al diseño del Backend de un proyecto de plataforma web, siendo el Backend la parte del proyecto que se preocupa del tratamiento, transmisión y recepción de los datos. Para aclarar los conceptos expuestos, una plataforma web se construye en base al Frontend, la parte que interactúa con el usuario u cliente, es el primer contacto con él y se puede entender como el aspecto visual o fachada

de la plataforma, es decir, lo que vemos cuando carga una página web, los lenguajes de programación usualmente utilizados para construir el esqueleto de la página web, su estilo (diseño) e interacción con el usuario son, respectivamente, HTML, CSS y JavaScript, estas tecnologías se encuentran del lado del cliente (Client-side). El Backend, por otro lado, es el aspecto interno de la plataforma y solo es accesible por programadores o administradores, en él se realiza la gestión y procesamiento de datos puesto que es el intermediario entre la base de datos y el Frontend, cada vez que se solicitan datos desde el Frontend (cuando un usuario interactúa con la plataforma web) se envían peticiones al Backend, el cual cuenta con credenciales para acceder a los datos y retorna los datos solicitados o una respuesta, las tecnologías utilizadas aquí, para gestionar y desarrollar el Backend se conocen como el lado del servidor (Server-side) y son comúnmente desarrolladas en JavaScript y Python. Cabe mencionar que el presente informe se centra mayoritariamente en el Backend del proyecto. La figura 2.1 muestra las tecnologías y lenguajes empleados en el desarrollo de la plataforma.

Figura 2.1: tecnologías para el lado del cliente y del servidor.



Fuente: elaboración propia utilizando contenido de https://fontawesome.com/license

Pese a que Django y Flask cumplen la función de estructurar el Backend, cada uno presenta ventajas sobre el otro, por ejemplo, Django automáticamente genera módulos básicos para trabajar en un proyecto, disponiendo de una gran cantidad de herramientas solo al inicializar el framework, pero estos módulos auto creados podrían no utilizarse haciendo el procesamiento de datos más lento, de todas formas Django presenta una estructura más amigable con el desarrollador que desconoce del área y es un excelente primer acercamiento pues cuenta con una excelente documentación. Flask, por otro lado, carece de módulos auto generados, y debe ser estructurado en su totalidad, desde cero, lo cual es una ventaja si se habla de desempeño pues solo contará con los módulos necesarios, pero en el caso de inexpertos presentará una complicación al carecer de una estructura definida y ordenada con la cual saber dónde está todo. Un ejemplo que contempla tanto Frontend como Backend es el inicio de sesión en una red social, el lector puede imaginarse un pequeño formulario escrito en HTML, se le aplica el diseño y normas gráficas de la marca, si debiese, con un archivo CSS, y se le añade funciones y efectos visuales con JavaScript, pero cuando se presiona el botón “iniciar sesión”, JavaScript envía una solicitud al servidor donde se encuentra la plataforma, el Backend ejecuta una función para comprobar las credenciales del formulario, realizando una consulta al gestor de base de datos y posteriormente enviando la respuesta al Frontend, ingresando a una nueva página personal con su sesión iniciada, o un error que da cuenta de credenciales erróneas, usualmente “Usuario o Contraseña inválidos”, la figura 2.2 muestra gráficamente la relación entre el Frontend y el Backend, la cual está en constante comunicación de solicitudes y

respuestas. Django y Flask no son los únicos framework y/o lenguajes de programación para ser empleados en estas tareas, pero son los que presentan mayor soporte por la comunidad que los utiliza, además de recibir constantes actualizaciones que son necesarios para mantener vivo el proyecto en el tiempo y la posibilidad de consultar y reportar errores directamente a los desarrolladores.

Figura 2.2: representación de comunicación entre el servidor y el cliente.

Fuente: https://www.cloudflare.com/learning/serverless/glossary/client-side-vs-server-side/

Cuando se trabaja en equipos de desarrollo en código de programación es normal tener inconvenientes respecto de cual archivo se encuentra en su última versión, cuales archivos fueron probados y cuales se encuentran en fase de prueba, además de ser capaz de comunicar los problemas que se provocan al realizar determinadas acciones en el código, estos problemas se resumen en el control de versiones, qué es determinar el estado de cada archivo y su responsable, y la plataforma GitHub soluciona esta problemática, facilita el control de versiones indicando cuando fue la última vez que se actualizó y por quien, además de permitir comunicar al equipo de trabajo y señalar el código que está generando problemas, ya sea porque no funciona, o no de la forma deseada. Los proyectos que se crean en GitHub son llamados repositorios y pueden ser públicos o privados, usualmente se utilizan repositorios públicos como curriculum de los trabajos que puede o ha realizado una persona, mientras que los repositorios privados son utilizados para proyectos

personales y/o de empresas para facilitar la comunicación de los desarrolladores. Todos los cambios que se realizan en el repositorio son almacenados y cada contribuidor puede verlos y restaurar el proyecto a un estado anterior si fuese necesario, una última ventaja que se debe destacar es que permite identificar conflictos del código, cuando múltiples personas lo están modificando.

Capítulo 3: Actividades realizadas

3.1 Capacitación

El periodo de practica inicia con la introducción al lenguaje de programación Python, el cual es ampliamente utilizado para realizar tareas de tratamiento de datos debido a su sintaxis explicativa, en general el nombre de una función es suficiente para saber lo que hace, es un lenguaje que es fácil de comprender para quienes inician en el mundo de la Data Science y cuenta con una comunidad muy extendida que atiende consultas de los usuarios, además de mantener actualizado el proyecto de libre acceso. La comunidad de Python ha desarrollado “librerías”, que son un conjunto herramientas para realizar un proceso determinado, dentro de ellas destaca Pandas, una librería orientada a la tarea de realizar Data Wrangling, lo que facilita el proceso de transformar los datos, Pandas internamente utiliza otra librería conocida como Numpy, la cual está optimizada para realizar operaciones con columnas y por ende reduce el tiempo necesario para aplicar transformaciones en los datos, para ilustrar el impacto de aplicar estas librerías, la figura 3.1, expone una prueba de desempeño con dos tipos de datos, enteros (*uint8*) y decimales (*float64*).

Figura 3.1: prueba de desempeño de la operación suma.

```
>>> x = numpy.ones((1000, 1000), dtype=numpy.uint8)
>>> %timeit x.sum(axis=0)
100 loops, best of 3: 2.36 ms per loop
>>> %timeit x.sum(axis=1)
1000 loops, best of 3: 1.9 ms per loop

>>> x = numpy.ones((1000, 1000), dtype=numpy.float64)
>>> %timeit x.sum(axis=0)
100 loops, best of 3: 2.04 ms per loop
>>> %timeit x.sum(axis=1)
1000 loops, best of 3: 1.89 ms per loop
```

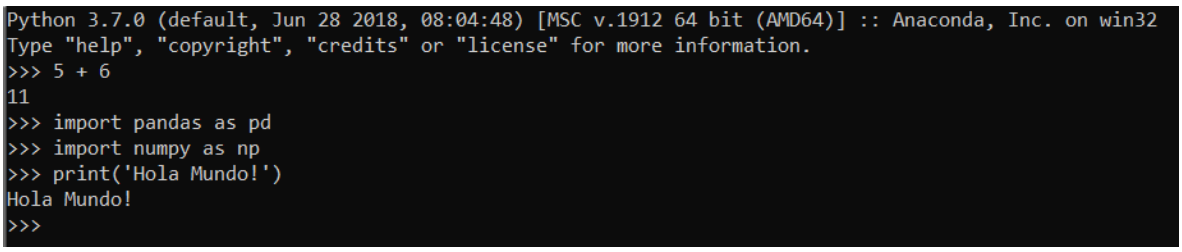
Fuente: https://stackoverflow.com/questions/17954990/performance-of-row-vs-column-operations-in-numpy

La prueba consiste en realizar la operación de suma en una matriz de dimensiones 1000 x 1000, donde cada elemento en la matriz tiene el valor 1, la operación se realiza sobre las filas (axis=0) y luego sobre las columnas (axis=1); la primera prueba, en ambos casos, realiza 100 iteraciones para la suma sobre las filas y luego 1000 iteraciones para la suma sobre las columnas. Puede apreciarse la elección de realizar operaciones sobre las columnas, considerando que en la actualidad se trabaja con Big Data y sus extensos conjuntos de datos.

Usualmente, al instalar un lenguaje de programación, se dispone de una línea de comandos similar a la que se muestra en la figura 3.2, en ella se pueden realizar diversas operaciones, por ejemplo, en la figura se muestra una suma y el ingreso de librerías Pandas y Numpy, hasta finalizar con el clásico ‘hola mundo’. En este punto se quiere resaltar que, no es imposible escribir un algoritmo mediante la línea de comando y guardarlo para poder ejecutarlo en otro momento o en otro equipo, pero requiere de más tiempo para realizarse y la tarea se complica cuando existe

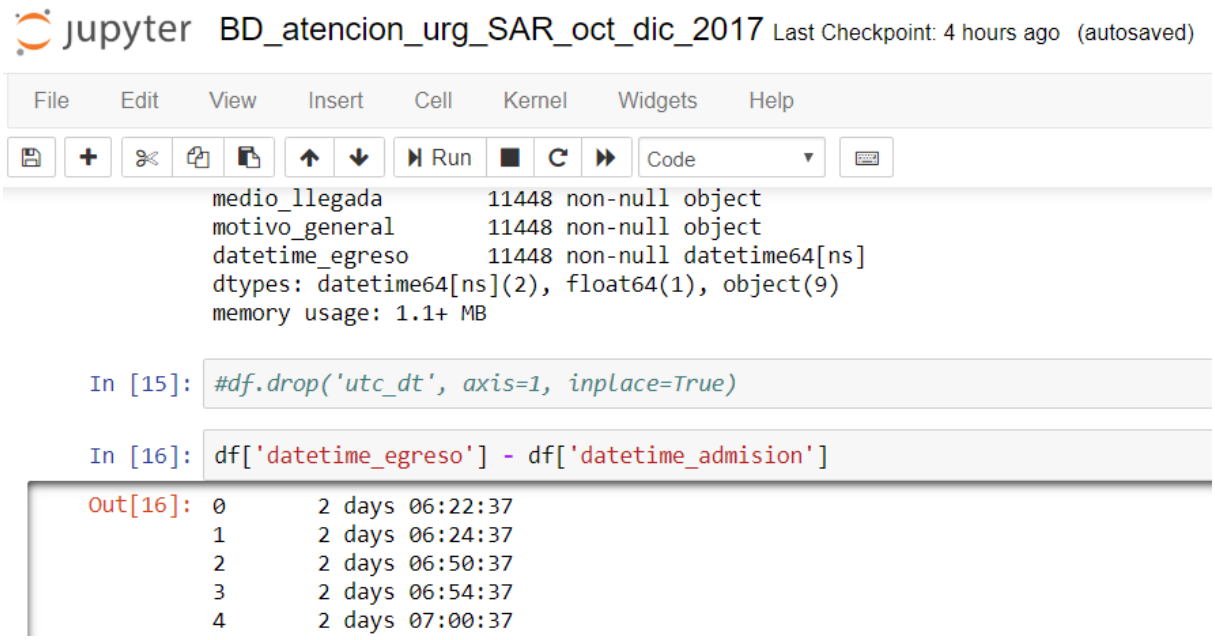
algún error en el código pues no es fácil identificarlo revisando el historial de comandos ejecutados, e incluso así, todo el código debe ser ejecutado nuevamente para volver al último estado funcional.

Figura 3.2: línea de comandos Windows, ambiente de Python.



Para solucionar este problema se cuenta con Jupyter notebook, una aplicación web que permite crear y compartir documentos que contienen código en vivo, es decir, se puede ver el resultado de la operación que se ejecuta, además de agregar texto para describir y/o contextualizar las operaciones que se realizan; es ampliamente utilizado en la transformación de datos, modelos estadísticos, visualización de datos y Machine Learning, entre sus usos más destacados, permite obtener una vista clara de la estructura del código y dividirlo de tal manera que es posible aislar un bloque de código que no funciona como se espera. La figura 3.3 muestra la aplicación Jupyter Notebook y un ejemplo de la ejecución de un bloque de código.

Figura 3.3: documento de Jupyter Notebook.



En la figura se puede visualizar la procedencia de los datos, siendo esta el SAR de Chiguayante en el periodo octubre a diciembre del año 2017, el código que se ejecuta *df['datetime_egreso'] – df['datetime_admision']* puede interpretarse como el tiempo en que estuvo un paciente siendo atendido (la diferencia entre el momento en que ingresa a admisión y el momento en que su ficha es devuelta a admisión e ingresada al sistema), el resultado de esta operación puede verse a continuación de la celda del código, por ejemplo, *2 days 06:22:37*, indicando que el paciente estuvo un total de 2 días, 6 horas, 22 minutos y 37 segundos. Cabe destacar que un Notebook de Jupyter se utiliza para ejecutar el algoritmo que está escrito dentro de él y entregar un resultado, pudiendo ser este un archivo en un formato determinado, usualmente *.XLSX*, *.CSV*, *.JSON*, o una conexión con el gestor de bases de datos para cargar estos datos en la base de datos.

Para los proyectos que se exponen en este trabajo se utiliza la tecnología de PostgreSQL cuando se refiere al motor de bases de datos, tiene la particularidad de trabajar con modelos relacionales de bases de datos como el **Modelo Entidad-Relación** (MER), que es una herramienta para representar y gestionar la posterior implementación de una base de datos relacional. El MER consta de tres aspectos principales: ***Entidades*** que representan a una tabla en la base de datos, una generalización de un objeto físico o abstracto que posee un conjunto de características; ***Atributos*** son las columnas presentes en cada tabla, un conjunto de características de una Entidad; ***Relaciones***, establecen la dependencia y conexión entre las Entidades.

Las relaciones entre Entidades explican cómo estas interactúan y dependen de otras, se distingue cuando una Entidad es fuerte, es decir, si esta no necesita de otra Entidad para existir, caso contrario, la Entidad presenta información detallada de un aspecto particular de la Entidad fuerte.

El Modelo Entidad-Relación presenta tres posibles Relaciones entre Entidades (se emplea el concepto de tabla indiferentemente con el de Entidad):

1:1: "*uno es a uno*", donde una fila[1] de una tabla solo se relaciona con otra fila de otra tabla. A modo de ejemplo, para las Entidades Automóviles y Conductores, un automóvil sólo puede tener un conductor y un conductor solo puede conducir su automóvil (haciendo la suposición, por ejemplo, de que son conductores de taxi y solo conducen su vehículo).

[1] En base de datos se emplea el concepto tupla, la cual es equivalente a una única fila de una tabla en particular.

1: n: "*uno es a n*[2]", donde una fila de una tabla se relaciona con múltiples filas de otra tabla. Para ejemplificar, las Entidades Automóviles y Dueños, un automóvil cuenta con un único dueño, mientras que un dueño podría contar con múltiples automóviles.

n: n: "*n es a n*", donde múltiples filas de una tabla, se relacionan con múltiples filas de otra tabla. Contando con las entidades Buses y Conductores, un bus puede ser conducido por múltiples conductores, de la misma manera un conductor puede conducir múltiples buses (contando con la suposición donde un conductor no puede ser dueño de un bus).

El anexo 2, presenta el esquema de base de datos del proyecto Open Tech en donde estos conceptos son aplicados.

Como **PostgreSQL** es el motor de la base de datos, la capa lógica que describe su funcionamiento y permite utilizar la base de datos, es necesario contar con un software adicional para administrar la base de datos y poder acceder a su contenido, **PgAdmin** es la herramienta que se utiliza para este propósito, añade seguridad al contenido que se almacena proporcionando credenciales a los usuarios que tienen permitido acceder y con ellas, se logra una conexión remota cuando se requiera acceder o cargar datos en la base de datos. La figura 3.4 muestra el código necesario para realizar una conexión remota con **PgAdmin** y la interfaz gráfica del gestor de bases de datos puede verse en el anexo 4.

[2] "*n*" representa un número superior a 1.

Figura 3.4: conexión con gestor de base de datos PgAdmin.

```
import psycopg2
from sqlalchemy import create_engine
from psycopg2.extensions import ISOLATION_LEVEL_AUTOCOMMIT

# Conexión a bd
local = {
    "database": "cis2analytics", # nombre db en postgreSQL
    "host": "localhost", # dirección host
    "password": "qazplmokn21", # contraseña de usuario postgreSQL
    "port": 5432, # puerto predeterminado de postgreSQL
    "user": "postgres" # usuario postgreSQL
}

database_connection = psycopg2.connect(**local) # genera conexión con db
cur = database_connection.cursor() # almacena la conexión

database_connection.set_isolation_level(ISOLATION_LEVEL_AUTOCOMMIT)

cur.execute("""
DROP SCHEMA IF EXISTS innovation CASCADE;
CREATE SCHEMA innovation AUTHORIZATION postgres;
""")

# Subir datos a DB
engine = create_engine("postgres+psycopg2://postgres:qazplmokn21@localhost:5432/cis2analytics")
df.to_sql(name='dataset', con=engine, schema='innovation', if_exists='replace', index=False)
```

En resumen, Python es un lenguaje de programación que cuenta con un alto soporte del proyecto, el cual le permite mantenerse actualizado y mantenido, posee una curva de aprendizaje que lo hace ideal para iniciarse en la Data Science y cuenta con las librerías necesarias para realizar la totalidad de operaciones requeridas para trabajar con datos, estas operaciones son extracción, transformación y carga; las principales librerías que se utilizan para realizar dichas operaciones sobre los datos son Pandas y Numpy puesto que cuentan con funciones que facilitan estas tareas y trabajan con la estructura de los datos de tal manera, que se optimizan las transformaciones que se deben hacer reduciendo el tiempo de ejecución. **Jupyter Notebook** es una aplicación que web que permite dividir el algoritmo de transformación de los datos en bloques, de esta forma el código es fácilmente aislado lo que permite identificar posibles errores en el diseño y posterior

implementación. Finalmente, se debe contar con datos estructurados, por ejemplo, según el Modelo Entidad-Relación pues permite integrar diferentes bases de datos, y accesibles remotamente, puesto que el objetivo del proyecto es generar una plataforma web, es necesario disponer de los datos cada vez que se ingrese a la plataforma, para ello el gestor de bases de datos **PgAdmin** proporciona las herramientas y seguridad necesarias para mantener la integridad de los datos.

3.2 Data Wrangling

Como se mencionó **Data Wrangling** es una metodología para realizar el proceso de transformación de los datos, que es la tarea más extensiva en tiempo del tratamiento de datos, su dificultad radica en lograr la estructura y formato adecuado para su posterior uso, a continuación, se mencionan problemáticas generales al realizar **Data Wrangling** y ejemplos del código utilizado para realizarlas.

Data Wrangling se divide en cuatro grandes tareas al tratar los datos, Transformar, Unir, Adaptar y Evaluar los datos utilizables, siendo la evaluación de las características o variables (columnas de datos en una tabla) la más importante, dependiendo del fin del estudio, existirán variables que podrían no ser necesarias, y por lo tanto deben eliminarse antes de realizar cualquier transformación en ellos, esta operación puede entenderse como un ahorro de tiempo de procesamiento, al igual como un ahorro de espacio de almacenamiento, por ejemplo, si al diseñar el esquema de la base de datos se cuenta con la variable edad y esta se almacena como un numero entero, obtener el promedio de la edad de las personas en la base de datos es una tarea trivial, mientras que si se almacena la fecha de nacimiento (considerando el formato *datetime* equivalente a dd/mm/aaa hh:mm:ss), se debe

extraer solo el año, realizar la diferencia con el año actual y convertir el formato de tipo texto, puesto que es el formato de origen de los datos, a numérico para poder realizar la operación de promedio. Usualmente la fuente de datos presenta un formato propio, determinado al momento de extraer o generar los datos, lo que podría generar problemas si no es el formato adecuado para un tipo de dato determinado como se demostró, ya sea una fuente local como un archivo Excel o remota como un sistema de almacenamiento de datos (Data Warehouse), una forma de tratarla es leyendo todo su contenido con el formato tipo texto, por ejemplo, los códigos del Sistema Armonizado de mercancías inician con la nomenclatura 0100 para animales vivos, si este dato fuese leído como número se habría convertido en 100, el cual no generaría ninguna coincidencia con su correspondiente producto al analizar los datos de importaciones o exportaciones.

Transformación de datos

La transformación de datos consiste en el proceso de corregir, homologar y borrar datos de una base de datos, cuando estos se encuentran incompletos y/o erróneos lo cual dificulta su visualización al no presentarse con claridad, además de generar problemas al momento de realizar análisis estadísticos y aplicar modelos de predicción, el objetivo de este proceso es diagnosticar el estado y calidad de los datos para posteriormente homogeneizarlos, por ejemplo, si se cuenta con datos numéricos de edades, es posible llenar datos faltantes con el promedio de las edades, lo cual no afecta esa medida de tendencia central, si se cuenta con información adicional como su ciudad de procedencia, se podría obtener la edad promedio de los habitantes de esa ciudad e ingresar los datos faltantes,

dependiendo del objetivo del estudio, la mejor alternativa podría ser dejar ese dato con el valor *null*, el cual no se considera cuando se realizan operaciones sobre los datos, claramente reemplazar los datos faltantes con el valor numérico 0 afectará significativamente las medidas de tendencia central.

Datos ruidosos son llamados aquellos que presentan una discrepancia significativa con el resto del conjunto de datos, también se debe de considerar el tratamiento de datos atípicos pudiendo existir dos enfoques, si se busca representar el comportamiento de una máquina que produce piezas en una cadena de producción, se deben de mantener los datos atípicos porque representan el comportamiento del sistema, pudiendo visualizar una falla del mismo (un error sistemático del equipo o del operador en un momento o turno determinado) y alertando de tal error al equipo de mantenimiento, o si se busca representar el comportamiento general, por ejemplo la cantidad de metros cuadrados de propiedades inmobiliarias, si el objetivo del estudio es comparar el indicador $\frac{UF}{m^2}$ en Departamentos y Casas, las ventas de terrenos en la base de datos se considerarían datos atípicos para el estudio. determinar si un dato es atípico, en datos numéricos. puede lograrse mediante la ecuación que se muestra en la figura 3.5, que corresponde a comprobar si un dato se encuentra en cierto intervalo, en este caso 4 desviaciones estándar, también se expone el código que crea la columna del indicador UF/m2

Figura 3.5: identificación de datos atípicos.

```
In [12]: df_['UF_M2'] = df_['uf']/df_['m2_l']

In [258]: df_ = df_[~((df_.m2_l-df_.m2_l.mean()).abs() > 4*df_.m2_l.std())]
```

Inconsistencias y discrepancias en los datos son tareas de la transformación de datos, la figura 3.6 presenta un claro ejemplo de un dato mal ingresado, no se puede hablar de un dato omitido porque el sistema asigna el valor por defecto *null* (Python asigna el valor *NaN* a estos casos, *not a number*) cuando no se ingresa.

Figura 3.6: datos de regiones en la Encuesta de Innovación en Empresas.

```
In [7]: df.region.unique()
Out[7]: array([1, 0, 2, 13, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 15, 14, '.'],
              dtype=object)
```

Los datos que se muestran corresponden a la Encuesta de Innovación en Empresas desde su quinta hasta su novena versión, puede observarse el signo de puntuación '.' y la región 0 dentro de la asignación de regiones, después de obtener más información de la fuente de los datos, se concluyó que la región 0 se utilizó para representar la agregación de varias versiones de la encuesta, mientras que el signo de puntuación es un error de ingreso de datos.

De la misma forma se presentan los diagnósticos de pacientes que asistieron al SAR, donde se destaca el diagnóstico de Constatación de Lesiones y Alcoholemia, nótese como se presenta en tres categorías diferentes, esto debido a que no están escritas exactamente igual, la función *unique()* en la primera línea del código se encarga de identificar todas las categorías únicas de la columna diagnóstico lo que ayuda a identificar y corregir el problema, esto se muestra en la figura 3.7.

Figura 3.7: categorías de diagnóstico SAR de Chiguayante.

```
In [15]: temp = df1.diagnostico.unique()
         temp = temp.tolist()
         temp.sort()
         temp
```

```
'CAIDA ,EMBARAZO DE 35 SEMANAS',
'CEFALEA CON BANDERA ROJA',
'CEFALEA FUERTE INTENSIDAD',
'CELULITIS PREORBITORIA IZQ',
'COLICO BILIAR',
'COLICO RENAL',
'COLOSTOMIA NO FUNCIONANTE',
'COMPROMISO DE CONCIENCIA EN ESTUDIO',
'CONSTATACION DE LESIONES',
'CONSTATACION DE LESIONES - VIF',
'CONSTATACION DE LESIONES Y ALCOLEMIA',
'CONSTATACION DE LESIONES+ALCOHOLEMIA',
'CONSTATACION DE LESIONES, ALCOHOLEMIA',
'CONSTATACIONES DE LESIONES',
'CONSTIPACION - FECALOMA',
'CONSUMO SUSTANCIAS',
'CONTSUION DE CRANEO MENOR DE 2 AÑOS',
'CONTUSION ANTEBRAZO',
'CONTUSION CERVICAL',
```

Al limpiar bases de datos que contienen textos es recomendable realizar un pre procesamiento con tareas estándar para resolver problemas comunes, por ejemplo, la diferenciación de palabras en mayúsculas y minúsculas, tildes que podrían estar o no incluidos en las mismas palabras lo cual genera múltiples categorías. Adicionalmente se muestra una función de utilidad cuando se realiza un pre procesamiento de los datos, borrar duplicados es una tarea esencial para disminuir trabajo innecesario junto con seleccionar las variables que efectivamente serán usadas en el estudio, la figura 3.8 muestra el comando que realiza tal acción; si se trata de datos numéricos se debe de tener en cuenta si existe dependencia lineal entre las variables, es decir, si una variable es producto de 2 o más variables en el conjunto de datos lo que afecta la aplicación de modelos de regresión lineal por ejemplo.

Figura 3.8: función para borrar datos duplicados drop_duplicates().

```
In [2]: #pre-procesamiento (general)

        #quitar duplicados
        df.drop_duplicates(inplace=True)

        #mostrar todas las columnas
        pd.set_option('display.max_columns', m)
```

Integrar bases de datos

Integrar bases de datos se refiere al proceso que combina múltiples fuentes de datos heterogéneas en una única estructura, puede imaginarse, por ejemplo, una empresa que cuenta con múltiples departamentos, digamos finanzas y operaciones, cuya integración de datos podría entregar información de cuanto actualmente se produce y de lo producido, cuanto se ha vendido, además de los ingresos que generaron, si se considera la base de datos de las sucursales con las que se cuenta, se podría contar con información suficiente para trasladar mercancía a sucursales que no tienen abasto y enfocar sus ventas de ciertos productos en ciertas sucursales, más radicalmente trasladar una fábrica a una localidad determinada. El potencial de lo que integrar bases de datos propone depende directamente de los datos que actualmente son capturados por la empresa, y de la tecnología con la que se disponga para obtener información y posterior conocimiento de ellos.

Adaptar datos

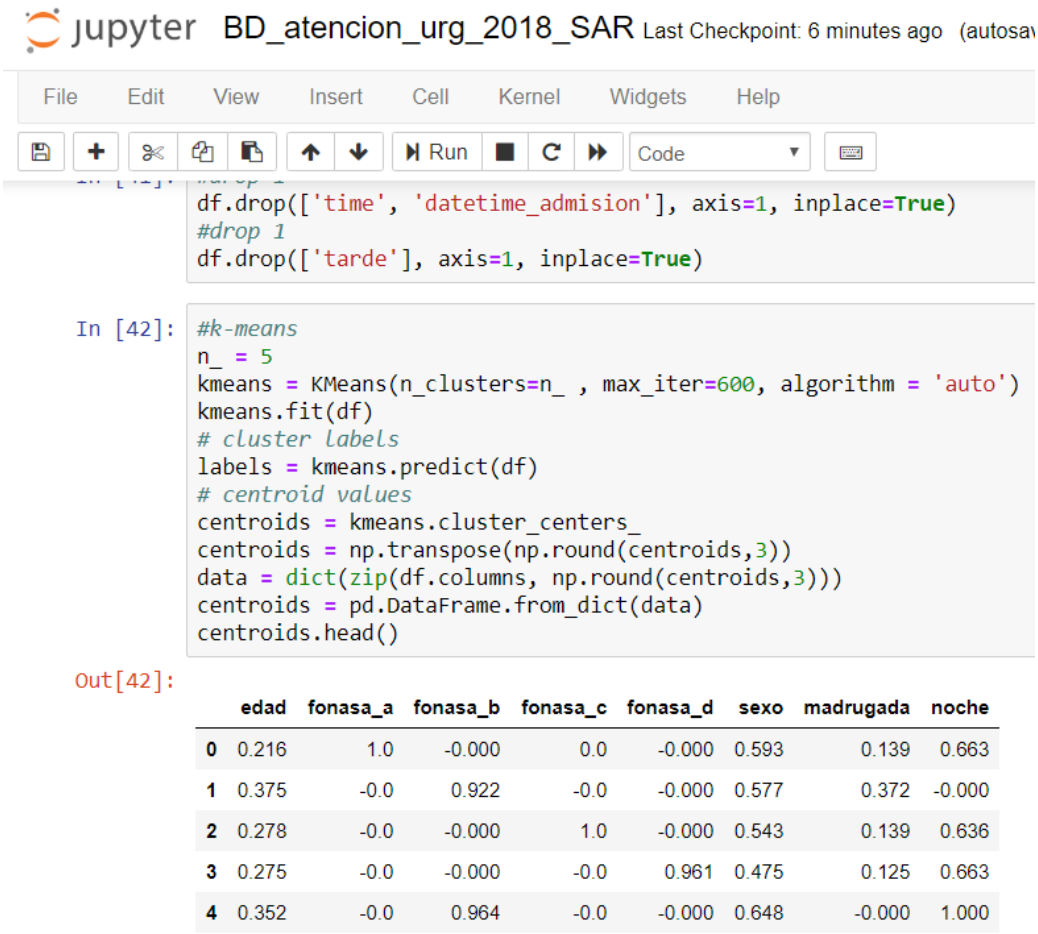
Adaptar datos considera todas las operaciones necesarias para transformar la estructura en la que se encuentran almacenados los datos, dichas operaciones dependen del uso posterior que se les dará a los datos; si el objetivo es visualizar los datos se debe de modificar su estructura y para ello se seguirán los pasos

necesarios para dejarlos en el formato Tidy previamente expuesto, es decir, las observaciones son filas únicas, se tiene una columna por variable y las tablas de datos son unidades observacionales (la tabla regiones contiene únicamente datos de cada una de las regiones, luego la tabla comunas solo contiene datos de las comunas y una región puede contener 1:n comunas, más una comuna pertenece únicamente a una región).Dependiendo de la estructura en que se presentan los datos, se deberán realizar operaciones adicionales de creación de columnas, si se tiene columnas con múltiples variables, eliminar, desagregar datos e incluso discretizar rangos de datos, todas las operaciones necesarias para llevar la estructura de datos al formato Tidy, se facilita la implementación modelos matemáticos y algoritmos de Machine Learning.

Si el trabajo inicial de transformación de datos es realizado correctamente la aplicación de algoritmos de clasificación o predicción se vuelve mucho más sencilla, un ejemplo de aplicación de un algoritmo de Machine Learning K-mean se puede observar en la figura 3.9 y 3.10, la librería de Python Sk-learn cuenta con variadas funciones que aplican estos algoritmos a una estructura determinada de datos, que es el objetivo de utilizar **Data Wrangling** y **Tidy Data**. K-means es un algoritmo de Machine Learning que tiene por objetivo generar clúster de datos con comportamiento homogéneo, es del tipo no supervisado, es decir no se conoce previamente la clasificación de las observaciones en un clúster determinado, estos algoritmos trabajan únicamente con datos numéricos lo cual necesita de una transformación adicional cuando se tiene datos de tipo texto, en general se tendrá tantas columnas adicionales como categorías tenga una variable.

Data Science es finalmente un conjunto de herramientas para realizar análisis de datos, aplicación de modelos matemáticos y estadísticos para identificar tendencias y patrones de comportamiento con el objetivo de ser utilizados en la toma decisiones, más un punto crítico de su aplicación es el origen de los datos, inicialmente por su veracidad y luego por su valoración, es distinto si un Ingeniero evalúa la calidad de los datos de fichas clínicas a un Médico, por lo tanto se debe contar con profesionales de distintas áreas que realicen un trabajo crítico al momento de diseñar la herramienta con tal de obtener información que efectivamente sea empleada para generar conocimiento.

Figura 3.9: modelo de Machine Learning, K-means.



	edad	fonasa_a	fonasa_b	fonasa_c	fonasa_d	sexo	madrugada	noche
0	0.216	1.0	-0.000	0.0	-0.000	0.593	0.139	0.663
1	0.375	-0.0	0.922	-0.0	-0.000	0.577	0.372	-0.000
2	0.278	-0.0	-0.000	1.0	-0.000	0.543	0.139	0.636
3	0.275	-0.0	-0.000	-0.0	0.961	0.475	0.125	0.663
4	0.352	-0.0	0.964	-0.0	-0.000	0.648	-0.000	1.000

Figura 3.10: procesamiento adicional para determinar cantidad de individuos en centroide.

In [43]:

```
def ClusterIndicesNumpy(clustNum, labels_array):
    return np.where(labels_array == clustNum)[0]

suma = 0
samples = []
per = []
for i in range(n_):
    samples.append(len(ClusterIndicesNumpy(i, kmeans.labels_)))
    suma = suma + int(len(ClusterIndicesNumpy(i, kmeans.labels_)))

for i in range(n_):
    per.append(round(samples[i]/suma, 3))

data = {'cantidad': samples,
        '%_total': per}

temp = pd.DataFrame(data)
centroids = centroids.join(temp)

centroids.head()
```

Out[43]:

	edad	fonasa_a	fonasa_b	fonasa_c	fonasa_d	sexo	madrugada	noche	cantidad	%_total
0	0.216	1.0	-0.000	0.0	-0.000	0.593	0.139	0.663	15438	0.298
1	0.375	-0.0	0.922	-0.0	-0.000	0.577	0.372	-0.000	6964	0.134
2	0.278	-0.0	-0.000	1.0	-0.000	0.543	0.139	0.636	8373	0.162
3	0.275	-0.0	-0.000	-0.0	0.961	0.475	0.125	0.663	10103	0.195
4	0.352	-0.0	0.964	-0.0	-0.000	0.648	-0.000	1.000	10959	0.211

El resultado que aparece como *output* del algoritmo es lo que se conoce como centroides, que son las características de cada grupo, por ejemplo, el grupo 1 en su mayoría son personas de mayor edad que los otros grupos, cuentan con previsión de salud Fonasa B, y asistieron al centro de salud durante la tarde, esta última categoría no se visualiza debido a que se utiliza como grupo de comparación. La cantidad de grupos generada es arbitraria y dependerá de quien realiza el análisis, no obstante, existen técnicas como el diagrama de codo que permite determinar un numero óptimo de centroides a los cuales asignar individuos con comportamiento similar. Un hecho importante a destacar es el grupo 0, que cuenta en su totalidad con pacientes de previsión Fonasa A, de los cuales aproximadamente 60% son

mujeres y que acuden al centro de salud durante la noche, lo que podría significar en destinar recursos a ese horario puesto que representa, 29,8% del total de atendidos, sin considerar el grupo 4 compuesto principalmente con pacientes de previsión Fonasa B, donde 64,8% son mujeres que se atienden únicamente por la noche.

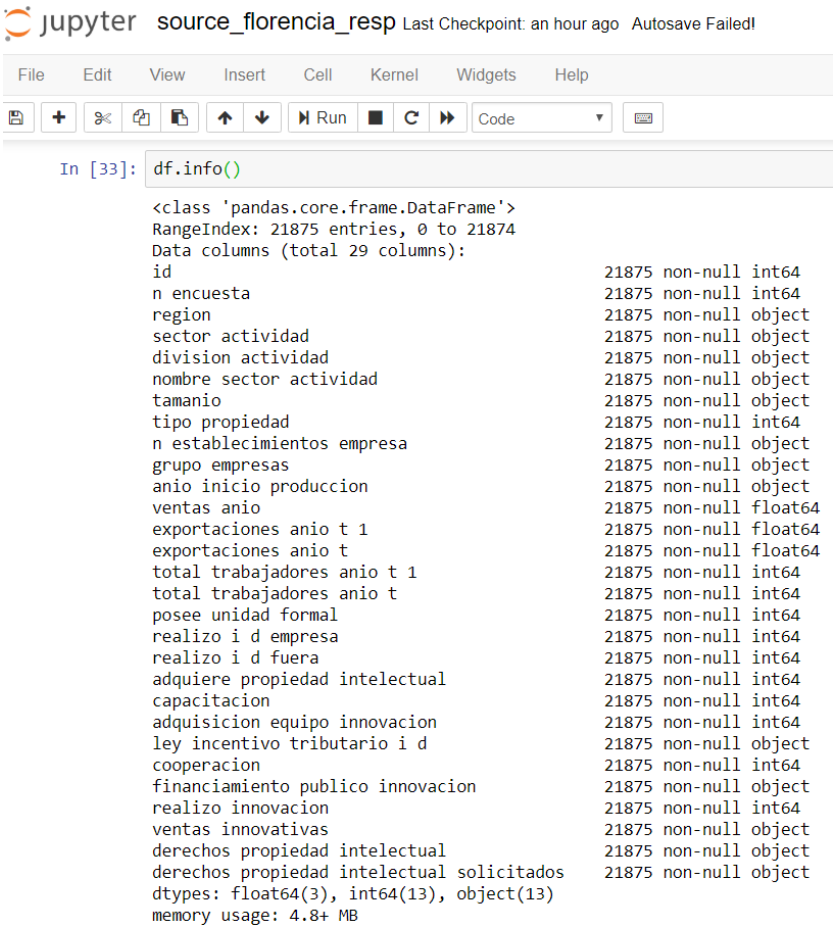
3.3 CIS2

El primer proyecto donde se aplican los conocimientos adquiridos de Data Science es la Plataforma de Innovación CIS2, la cual tiene el objetivo de exponer al público información clara de la estructura y de la capacidad de generar innovación de las empresas en Chile, en la Encuesta de Innovación se responde a preguntas como si se cuenta con derechos de propiedad intelectual, si realiza investigación y desarrollo, y si tiene una unidad formal en la empresa que lo desarrolle, esto agrupado por sector productivo y región del país. En la Encuesta se tiene la responsabilidad de transformar y cargar los datos en la base de datos, por lo que se cuenta con las bases de datos de las Encuestas de Innovación (desde la quinta hasta la novena versión) en formato de archivo Excel.

Inicialmente se determina con cuales columnas de la base de datos se trabajará y el estado de estas, existe el escenario donde una columna cuente con los datos buscados, pero que esta se encuentre prácticamente vacía y/o con datos de baja calidad, en el anexo 6 se muestra parte de la base de datos con la que se trabajó, una buena práctica puede apreciarse en dicha base de datos y es que cuenta con una hoja adicional que entrega la definición de los códigos con los que se nombra las columnas, en el anexo 7 se encuentran las columnas presentes en la base de

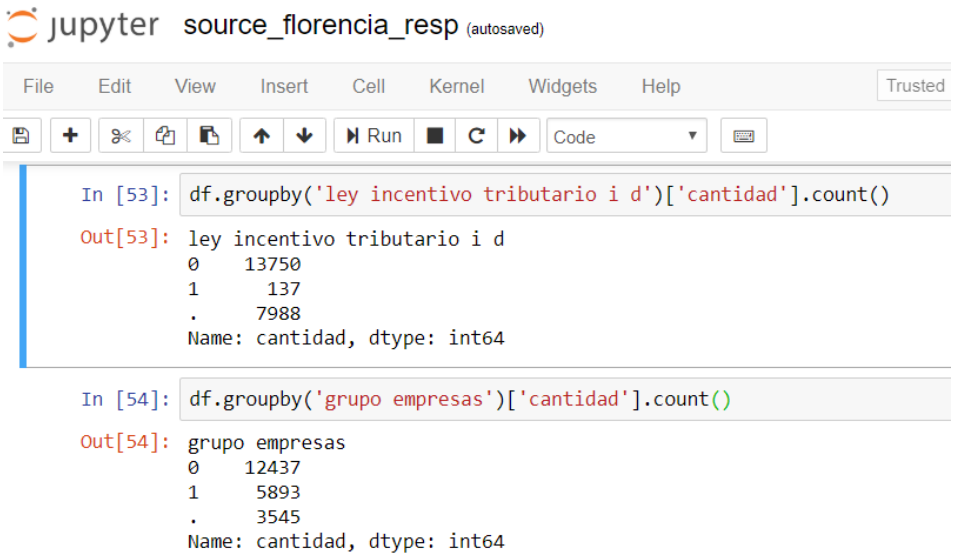
datos, sus códigos, su definición, y posibles campos de respuesta si hubiese, como las opciones de selección múltiple. Posterior a la selección de variables con las que se trabajará, se procede a visualizar indicadores de estructurar de datos, por ejemplo, de las columnas seleccionadas para trabajar cuantas filas están completas, y de ellas cuantos datos son válidos, en la figura 3.11 se ve el resultado del comando *.info()* que retorna todas las columnas del archivo Excel, el tipo de dato que almacena (*int64* corresponde a datos numéricos enteros, *object* datos de tipo texto y *float64* datos numéricos que contienen decimales) y la cantidad de datos no nulos con los que se cuenta.

Figura 3.11: resultado retornado por la función .info().



Muchas veces presenta información general respecto de los datos con los que se trata, pero no se debe de guiar únicamente por esto, pese a que se muestra la cantidad de datos no nulos, no dice nada del contenido; similar al objetivo de la función *.unique()*, se puede determinar la cantidad de coincidencias para una categoría determinada agrupando los datos en una columna particular, la figura 3.12 muestra el agrupamiento para dos columnas particulares, “ley incentivo tributario i d” que hace referencia a si una empresa hizo uso de la ley de incentivo tributario para aplicar innovación y desarrollo, y “grupo empresas” que hace alusión a si una empresa pertenece o no, a un grupo de empresas.

Figura 3.12: ejemplo de distribución de datos por categorías.



Se retornan las categorías 0, 1 y ‘.’, además de la cantidad de coincidencias, idealmente esta columna tendría una categorización binaria, 1 si hizo uso de la ley mencionada y 0 si no, pero los campos donde no se indicó una alternativa, se rellenó con el signo de puntuación ‘.’, entonces ya no se cuenta con 21875 datos válidos,

ahora se tiene 13887, sin olvidar el hecho que muy pocas empresas han utilizado la ley de incentivo tributario.

Respecto de la toma de decisiones para solucionar este problema, es preferible convertir este signo de puntuación al valor *not a number* (NaN de Python), que el gestor de base de datos interpreta como *null* (valor nulo) al momento de ser cargado en una base de datos, de esta manera no se muestra como categoría en una posterior visualización y desde el punto de vista de la optimización, reduce la cantidad de espacio que se necesita para almacenar ese dato en dicha base de datos.

Otro punto importante es el tratamiento de datos tipo texto, tienen la particularidad de tener diferentes caracteres, no solo alfanuméricos, y esto presenta una problemática al transformar los datos, por ejemplo, hay caracteres que no existen en el alfabeto estadounidense, pero si en el nuestro, en la figura 3.13 puede verse un ejemplo de un usuario en la base de datos, aquí el carácter *ñ* es codificado.

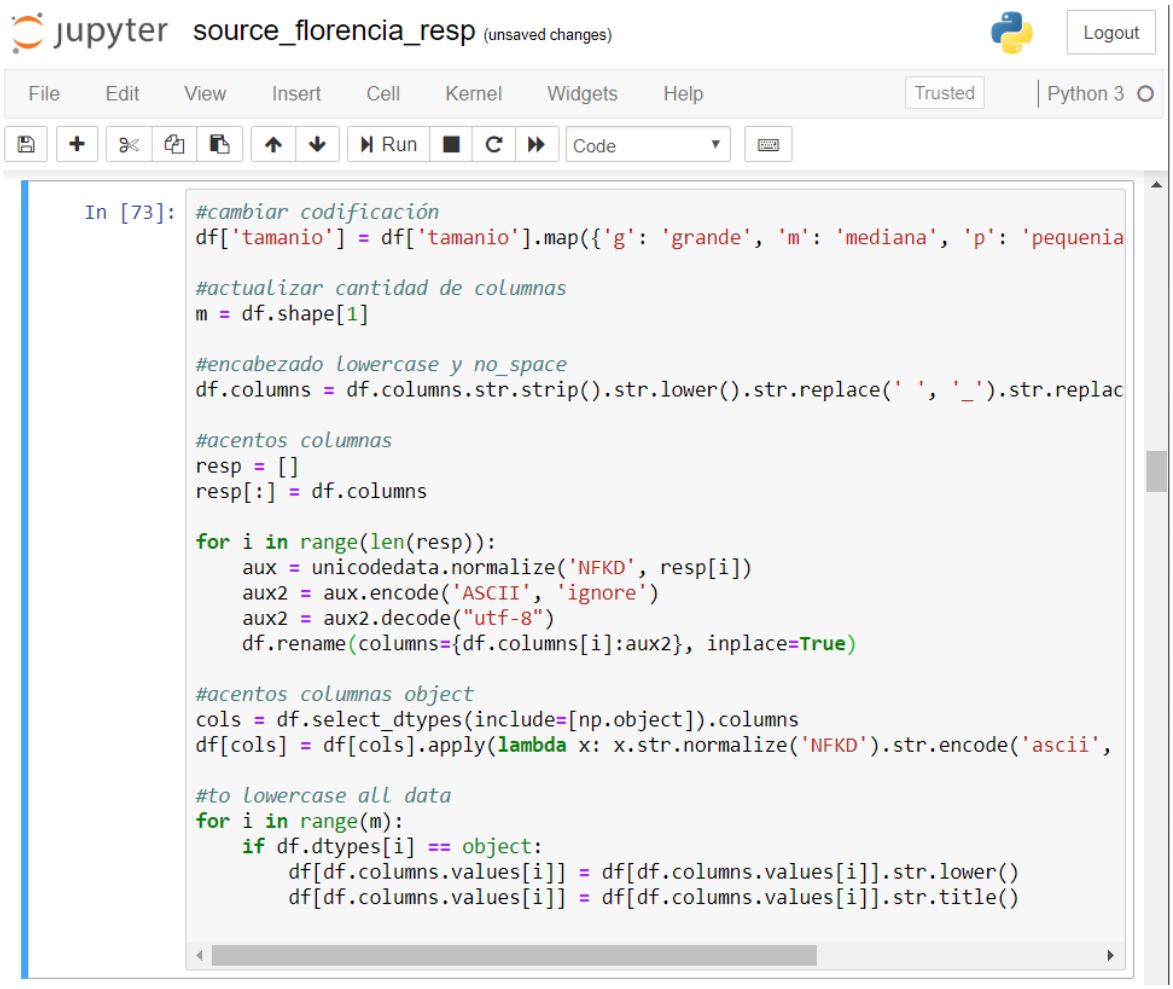
Figura 3.13: codificación de caracteres en base de datos.


```
{
    "model": "common.persona",
    "pk": 999,
    "fields": {
        "created": "2019-01-10T11:59:03.813Z",
        "modified": "2019-01-10T11:59:03.813Z",
        "firstname": "Ignacio",
        "lastname": "Mu\u00f1oz",
        "rut": "",
        "phone": 999999999,
        "email": "",
        "linkedin": ""
    }
}
```

El tratamiento de estos casos equivale a convertir los caracteres especiales a un formato estándar, de esta manera se puede decodificar el texto y mantener su interpretación, evitando problemas posteriores donde aparecen símbolos ilegibles; la figura previamente expuesta está codificada en un formato Unicode, luego, al momento de decodificar es legible para el idioma de destino.

Una parte del script se muestra en la figura 3.14, transforma los caracteres a Unicode, los nombres de las columnas a minúscula, y las columnas con datos de tipo texto son transformadas a minúsculas y posteriormente vuelve mayúscula la primera letra de cada palabra, el trabajo se traduce en generar la expresión que se muestra en el script para solucionar dicho problema, pero con la capacidad de ser reutilizado en posteriores trabajos.

Figura 3.14: automatización de transformación de caracteres especiales.



Un último aspecto a mencionar es la transformación de datos categóricos, en la figura 3.14 se pudo ver la transformación de caracteres especiales, pero que mantienen su significado, en la figura 3.15 se puede ver la selección de la columna “tamanio”, que hace referencia al tamaño de la empresa con las categorías “g”, “m” y “p”, puesto que el objetivo final es la visualización de estos datos, es más comprensible si se presenta de otra forma, en este caso es posible inferir a que corresponde dicha forma, además de corroborarlo con la fuente de los datos.

Figura 3.15: conversión de categorías.

```
In [73]: #cambiar codificación
         df['tamanio'] = df['tamanio'].map({'g': 'grande', 'm': 'mediana', 'p': 'pequenia
```

Dentro del informe el termino script será utilizado para referirse al algoritmo que realiza el trabajo de obtener datos de una fuente de datos, transformarla y posteriormente cargarla en la base de datos, el trabajo que se realiza en este proyecto viene siendo generar un script tal que realice las operaciones listadas en cualquier otro entorno, de esta forma quien administra el Backend y la correspondiente base de datos del proyecto puede ejecutarlo y contar con todos los datos disponibles.

3.4 Open Tech Biobío

Open Tech Biobío es una iniciativa de tres entidades del ecosistema innovador de la región, Casa W, IRADE y Corporación Desarrolla Biobío, cuyo objetivo es generar vínculos entre empresas y emprendedores que aportan soluciones innovadoras, el proyecto en cuestión se divide en hitos entregables que contemplan, un informe que contiene la estructura de la base de datos (Modelo Entidad-Relación), una página web para inscribirse en los distintos eventos, una plataforma web que muestre detalles del evento, los sectores de actividades económicas y sus representantes, galería fotográfica de eventos anteriores, noticias y próximos eventos; y finalmente, un panel de administración con indicadores para representar el impacto que genera el evento en sus asistentes.

En el proyecto Open Tech Biobío se aplican los conocimientos de Backend, es decir, se crea un servidor que gestiona las solicitudes del Frontend, y se utiliza el gestor

de bases de datos, contando con las credenciales necesarias para realizar consultas a la base de datos, este punto es esencial pues es el núcleo de la disponibilidad de datos, si se quisiese saber cuántas vinculaciones se dieron en el evento, diferenciando los sectores económicos y el tipo de vinculación que se dio, pudiendo ser vinculación efectiva, si esta se realizó en el evento, correo, teléfono y perfil de LinkedIn, si dichos datos fueron solicitados en la plataforma.

Un proyecto de Backend, según la estructura de Django, cuenta con un modelo inicial como el que se muestra en la figura 3.16, nótese que esta estructura corresponde a carpetas y archivos con la extensión *.py*, de la misma forma que un documento de MS Word tiene la extensión *.docx*, los archivos de código Python poseen la extensión *.py*; se tiene la carpeta *mysite* y dentro de ellas el archivo *manage.py* y otra carpeta con el mismo nombre *mysite* con otros archivos dentro.

Figura 3.16: estructura básica de proyecto en Django.

```
mysite/
    manage.py
    mysite/
        __init__.py
        settings.py
        urls.py
        wsgi.py
```

Fuente: https://docs.djangoproject.com/en/2.1/intro/tutorial01/.

Dentro del proyecto *mysite* se crea la aplicación *polls* y junto con ella las carpetas y archivos que se muestran en la figura 3.17.

Figura 3.17: estructura básica de aplicación en Django.

Fuente: https://docs.djangoproject.com/en/2.1/intro/tutorial01/.

Las figuras previamente expuestas corresponden a un proyecto de ejemplo para introducir el framework Django, para desarrollar el Backend de un proyecto de plataforma web Django cuenta con tres procesos esenciales que ocurren en tres diferentes archivos, *urls.py, views.py* y *models.py*; *urls.py,* con los cuales se administra todo el Backend de la plataforma, en la figura 3.18 se presenta un diagrama de flujo de los procesos que ocurren en dichos archivos, si se toma el ejemplo de un formulario, una vez ingresados los datos y validados por el Fronend (la validación puede ser tan sencilla como requerir que todos los campos sean llenados, hasta solicitar que los datos ingresados tengan un formato especifico, como que el correo efectivamente cuente con un dominio, por ejemplo @dominio.org o que el dígito verificador ingresado en el campo RUT sea el que corresponde según el algoritmo diseñado para ello), el usuario hace *click* en un botón de ingresar y el Frontend realiza una solicitud mediante los métodos POST o GET hacia un enlace visible para el usuario si se realiza por el método GET o internamente si se realiza con el método POST; los métodos POST y GET son protocolos para comunicar clientes con servidores, el método GET usualmente es

utilizado para solicitar datos, por ejemplo los siguientes enlaces son solicitudes GET porque es posible “ver” datos directamente en el enlace, nótese como cambia el número de la página web de 2 a 3 como argumento en la solicitud, el enlace es una búsqueda en la página web de GitHub por la palabra “tidy” en los repositorios:

https://github.com/search?p=2&q=tidy&type=Repositories

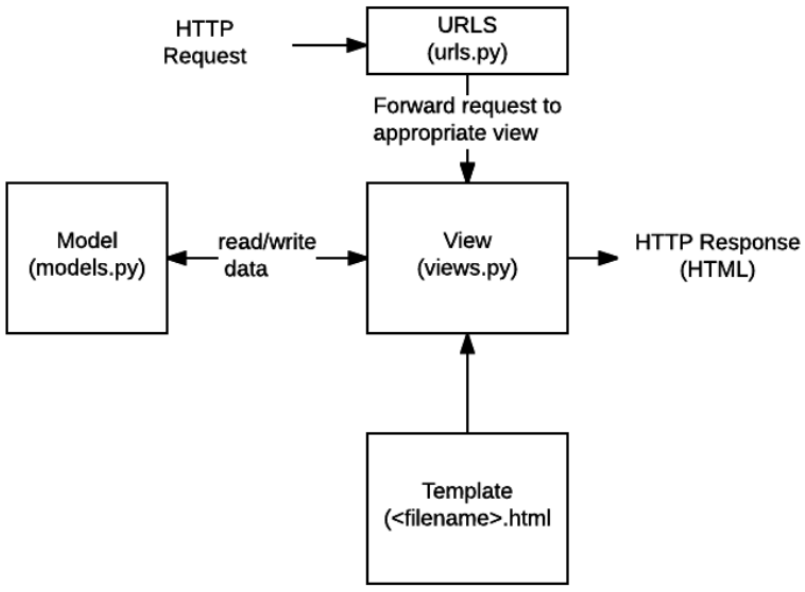
https://github.com/search?p=3&q=tidy&type=Repositories

No obstante, existen datos sensibles que no se deben mostrar a todo el público como contraseñas, las cuales se envían a la base de datos mediante el método POST, si hipotéticamente se realiza un inicio de sesión en GitHub en el enlace https://github.com/login, el próximo enlace visible es https://github.com/, el mismo ejercicio puede realizarse con plataformas de redes sociales, internamente se envía un mensaje al servidor indicando que se realiza una solicitud con el método POST.

Un usuario ingresará sus datos en el formulario y el Frontend realizará una solicitud en un enlace particular que está contenido en el archivo *urls.py*, si el enlace y el contenido son válidos (como medida de seguridad el Backend también cuenta con un formulario interno para validar los datos que se reciben) se accede al archivo *views.py* que contiene un conjunto de comandos en respuesta a dicha solicitud, si consideramos un formulario con los campos RUT, nombre y dirección, el archivo models.py también contiene esta estructura, los archivos *models.py* contienen el modelo de una base de datos, que no es otra cosa que la traducción de un Modelo Entidad-Relación al lenguaje de programación de la base de datos, de esta forma se realiza una validación sobre si se recibieron todos los campos del Frontend y en

el formato que está indicado en el modelo; manteniendo el ejemplo del formulario, *views.py* recibirá los datos ingresados y realizará operaciones dependiendo del contenido y solicitud, si suponemos que se ingresó un RUT, se comprobará mediante una solicitud a la base de datos si ya existe y retornará al usuario el mensaje “ya te encuentras registrado”, de otra manera podría actualizar los datos, indicar que el registro fue exitoso, o alertar al usuario que no se ha ingresado un RUT válido, si el RUT fue encontrado, se procede a solicitar al gestor de base de datos que lo almacene en la base de datos, se realiza otra solicitud con las credenciales del gestor de bases de datos y si son válidas se procede a guardar el registro. Al programar la interacción con usuarios se debe de pensar en todas las posibles solicitudes y respuestas necesarias para orientar las acciones del usuario.

Figura 3.18: Backend en Django.



Fuente: https://developer.mozilla.org/es/docs/Learn/Server-side/Django/Introducci%C3%B3n.

En la figura 3.19 se muestra el archivo urls.py del proyecto Open Tech que contiene algunos enlaces válidos para el Backend, se puede distinguir cuales utilizan métodos POST, como api/inscripción y api/vinculación, y métodos GET como api/personas/<id>, este formato representa que el enlace recibirá un parámetro adicional <id> que es necesario para ejecutar las funciones que están programadas en *views.py*.

Figura 3.19: archivo urls.py.

```
from django.urls import include, path

import api.views
import eventos.api
from opentech.admin.views import home_metadata
from opentech.images.views import galeria_home

urlpatterns = [
    #personas
    path('api/inscripcion', eventos.api.inscripcion, name='inscripcion'),
    path('api/personas/<id>', api.views.export_data_personas, name='personas'),
    path('api/personas/delete/<id>', api.views.delete_personas_id, name='delete'),
    path('api/personas/update/<id1>/<id2>', api.views.update_personas_evento_id, name='asistencia'),

    #vinculaciones
    path('api/vinculaciones/<id>', api.views.vinculaciones, name='vinculaciones_efectivas'),
    path('api/network/<id>', api.views.vinculacion_personas, name='vinculaciones_'),
    path('api/vinculacion/', api.views.vinculacion_usuarios, name='vinculacion_usuarios'),
```

Ya se mencionó que el archivo *urls.py* se comunica con *views.py*, en la figura anterior el enlace api/personas/update/<id1>/<id2> hace referencia a una función en el archivo *views.py*, el cual tiene por objetivo actualizar el estado de asistente al evento de la persona <id1> en el evento <id2>, de esta manera se llama a la función que se muestra en la figura 3.20.

Figura 3.20: ejemplo de api en archivo views.py.

```python
@csrf_exempt
def update_personas_evento_id(request, id1, id2):
    temp = Asistencia.objects.get(evento_id=id1, pk=id2)
    temp.confirmado = True
    temp.save()
    response = temp.as_dict()
    return JsonResponse(response)
```

El aspecto esencial del Backend es la definición del modelo y sus relaciones, tal como el Modelo Entidad-Relación, tener un modelo claro y bien definido permite responder a preguntas complejas respecto de los datos con los que se cuenta, en el archivo *models.py* se almacena la estructura de todas las tablas, sus relaciones con otras tablas y el tipo de datos que almacena (sus columnas en la base de datos), un extracto de dicho archivo se puede ver en la figura 3.21.

Figura 3.21: ejemplo de modelo en archivo models.py.

```python
class Asistencia(models.Model):
    created = models.DateTimeField(auto_now_add=True)
    modified = models.DateTimeField(auto_now=True)
    evento = models.ForeignKey(Evento, models.CASCADE)
    persona = models.ForeignKey(Persona, models.CASCADE)
    organizacion = models.ForeignKey(Organizacion, models.CASCADE)
    cargo = models.CharField(max_length=255, blank=True)
    confirmado = models.BooleanField(default=False)
    ip = models.CharField(max_length=255, blank=True)
    browser = models.CharField(max_length=255, blank=True)
    os = models.CharField(max_length=255, blank=True)
    device = models.CharField(max_length=255, blank=True)
    hash = models.CharField(max_length=255)
```

Las principales tareas desarrolladas para el proyecto Open Tech fueron el desarrollo del Backend y la gestión de la base de datos, y dentro de ellas generar *endpoints*, que son datos filtrados y procesados para obtener información y ser utilizados para generar conocimiento, estos *endpoints* se muestran como indicadores y visualizaciones que representan lo que fue el evento, se pueden encontrar en el

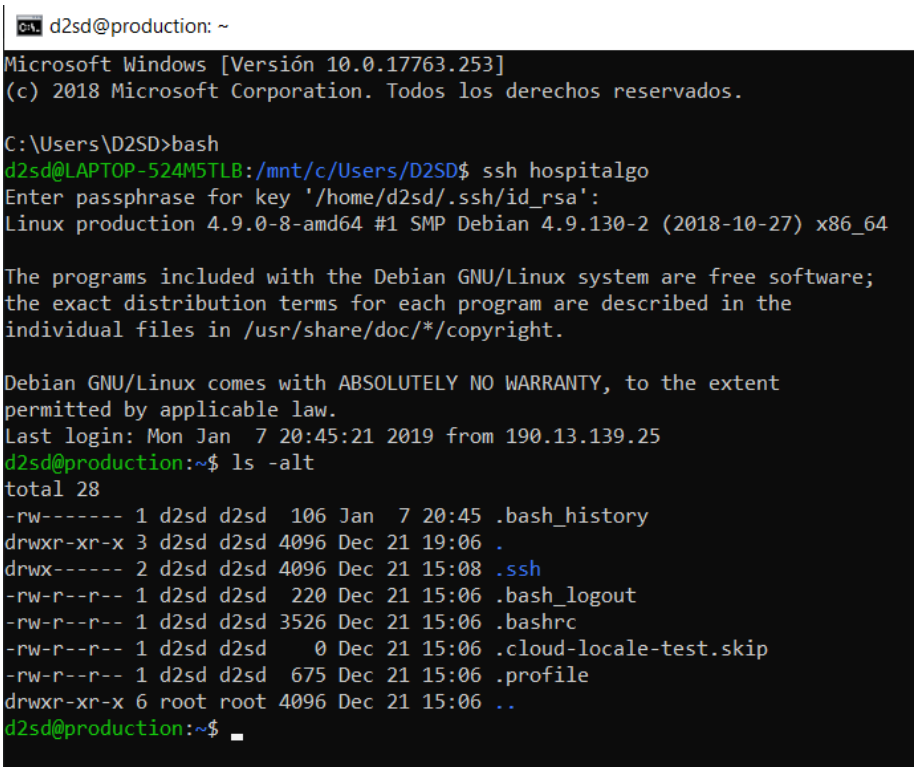
panel de administración de Open Tech y en la página principal que destaca los sectores productivos y sus correspondientes indicadores, asistencia, inscripciones, vinculaciones y un grafo de redes agregados por sector económico.

3.5 HOSPITAL GO

HOSPITAL GO es un proyecto del área de la salud, su objetivo es entregar información de diagnósticos y concurrencia de pacientes al alto mando del DAS y SAR de Chiguayante, para tomar decisiones basadas en evidencia que mejoren su calidad de atención, disponiendo de recursos humanos e insumos médicos para tratar a los pacientes y mejorar su calidad de vida cuando se les necesite. En HOSPITAL GO se utiliza todo el conocimiento adquirido de los proyectos anteriores, Data Wrangling para tratar los datos suministrados por el SAR, y posteriormente Data Science aplicando técnicas de Machine Learning para generar subconjuntos de datos con características similares, se utiliza el algoritmo de K-means con los datos suministrados, luego se desarrolla el Backend del proyecto y se adquiere la responsabilidad de operar el servidor remoto del proyecto, este último aspecto es nuevo y consiste en administrar el almacenamiento físico del proyecto; en el proyecto Open Tech se desarrolló todo el código necesario para hacerlo funcionar, pero más tarde este código fue trasladado a un servidor físico para mantenerlo disponible 24/7, es decir, se arrienda un servidor, que es un disco duro con un sistema operativo instalado, almacenado en algún lugar del mundo y que se mantiene encendido para recibir solicitudes remotamente, el acceso a dicho servidor presenta mayor seguridad y se realiza mediante las credenciales de una clave pública y una clave privada.

Secure Shell (SSH), es un protocolo de seguridad que permite a un usuario acceder y controlar un servidor remoto, este protocolo tiene la particularidad de cifrar la transferencia de datos entre el servidor y el usuario, se utiliza un cifrado asimétrico que utiliza dos claves, una pública y una privada; la clave pública es utilizada para cifrar un mensaje en el origen, mientras que la clave privada es utilizada para descifrar el mensaje en el destino, solo este par de llaves puede obtener el contenido del mensaje. Para poder utilizar este protocolo se debe de generar el par de claves pública/privada, la clave pública es entregada a quien administra el servidor para ser almacenada como usuario admitido y quien posee la clave privada procede a intentar la conexión con el servidor remoto, cuando se realiza la conexión por primera vez se solicitará generar credenciales adicionales, posteriormente solo se necesita del comando que se muestra en la figura 3.22.

Figura 3.22: conexión exitosa con servidor remoto.



El objetivo final de este aspecto es poder transferir los datos del proyecto, el Backend, el gestor de la base de datos y la misma base de datos, a un servidor remoto que mantenga disponible la plataforma web; una vez establecida la conexión es posible realizar esta transferencia de datos y ejecutar el proyecto remotamente.

La plataforma HOSPITAL GO, se presenta como un constructor que dispone al usuario con los datos procesados, de esta manera él puede estructurar la visualización según su criterio, no obstante, cuenta con sugerencias de gráficos y variables que utilizar en ella. Le proyecto contó con un Médico por parte del SAR de Chiguayante el cual presentó *feedback* respecto de que datos son útiles para él, además del director del DAS de Chiguayante que requirió iteraciones adicionales, como se mencionó fue necesario contar con un miembro en el equipo que tenga conocimiento del sector para valorar el trabajo realizado y como este le ayuda a visualizar la situación y periodos críticos del centro asistencial.

Capítulo 4: Resultados y Reflexión

Para cada proyecto mencionado en este informe se logró un producto final que fue entregado a los solicitantes y cada plataforma en cuestión posee un enlace disponible para ser visitado con la excepción de HOSPITAL GO que necesita de credenciales para acceder a su contenido.

Para el caso de CIS2, el script para generar el procesamiento de los datos se puede encontrar en el siguiente enlace: https://github.com/D2SD/practica_tutelada/tree/master/cis2, además, el producto final puede visitarse en https://innovacion.cis2.io/.

El proyecto Open Tech Biobío puede encontrarse en el enlace: https://www.opentechbiobio.cl/, así también el código fuente del Backend en https://github.com/D2SD/practica_tutelada/tree/master/opentech.

Finalmente el enlace para el proyecto HOSPITAL GO es el siguiente https://hospitalgo.danalytics.app/, las credenciales son solicitadas para acceder a la página principal, pero el código fuente del proyecto se encuentra en el siguiente enlace https://github.com/D2SD/practica_tutelada/tree/master/hospitalgo.

Los códigos expuestos en la plataforma GitHub no cuentan con las bases de datos con las que se trabajó, el objetivo de exponer el script es dimensionar el tiempo de trabajo que se requirió para desarrollar el producto final y el proceso de aprendizaje, y para ello GitHub permite visualizar la cantidad de líneas de código que se crearon, así también las que fueron eliminadas y el periodo de tiempo en el cual se trabajó.

Los proyectos planteados presentaron requerimientos únicos, así también como la comunicación con las personas que los solicitaron, con algunos fue más fácil tratar puesto que entendían en aspectos generales, el trabajo que requería cada proyecto y las tecnologías que se utilizarían, mientras que con otros se necesitó de un mayor esfuerzo para entender el objetivo de cada proyecto y dimensionar su dificultad. Cada proyecto conllevó un debido aprendizaje, no solo del aspecto técnico, necesario para desarrollar las plataformas expuestas como los lenguajes de programación necesarios, tecnologías y tendencias actuales sobre la implementación de las mismas, si no también habilidades para comunicarse con equipos multidisciplinarios, Diseñadores, Ingenieros Civiles Informáticos y otros Ingenieros Civiles Industriales que también desarrollan la tarea de programar y ven como un nuevo mundo de posibilidades se presenta, puesto que cuentan con las habilidades blandas para dialogar y entender los requerimientos del proyecto por quienes los solicitan, determinar si es factible, y cuantificar los requerimientos para lograrlo. Cada vez que se realiza Data Science, es necesario contar con un integrante del equipo que posea conocimiento del área en la que se trabaja y sea capaz de identificar cuales variables de los datos son necesarias y en qué forma, si se toma como ejemplo HOSPITAL GO, fue necesario contar con un Médico que clasificara e interpretara los diagnósticos realizados por sus pares, de esta forma la información se presentará más claramente a quienes toman las decisiones de presupuesto y recursos humanos asignados al centro asistencial de urgencia. Saber dimensionar la magnitud de un proyecto, su alcance y limitaciones, es una tarea pendiente que genera problemas cuando no se habla el mismo idioma, por eso se considera necesario contar con recursos humanos capacitados en tecnologías

actuales por parte del contratante para mejorar la fluidez del proyecto en cuanto a requerimientos y objetivos planteados.

Es un mundo particular el de la programación, puesto que quienes se encuentran en él son reconocidos por los proyectos en los que han participado más que por su origen o títulos, no es que un título universitario en informática o ciencias de la computación no tenga valor, pero no demuestra de lo que es capaz un profesional, se busca como un complemento para quienes ya cuentan con la experiencia y aptitudes necesarias para persistir como autodidactas, que es bastante común en este mundo. Considerando el escenario de la revolución digital y el impulso de la innovación en la industria, un Ingeniero Civil Industrial puede ubicarse en el punto central del proyecto porque es capaz de comunicarse efectivamente con el cliente para determinar sus necesidades de tecnologías de la información y traducir estos requerimientos al aspecto técnico para ser desarrollados e incluso participar de este desarrollo durante todo el proceso. Chile tiene una necesidad de revolución digital, pero no por falta de tecnología, sino de profesionales, recursos humanos, que sean capaces de utilizar la tecnología con la que se cuenta y adaptarla para solucionar problemas en áreas prioritarias, es necesario llevar esta digitalización a quienes la requieren. Dentro de las áreas prioritarias destaca la salud, y el hecho de que aún se necesite contar con registros físicos es alarmante, aunque es posible apreciar ambos extremos, durante el trabajo de campo del proyecto de HOSPITAL GO se dirigió a distintos centros de salud que simplemente no contaban con proveedores de servicio de ficha clínica digital y quienes contaban con más de un servicio, en este último caso la empresa que proveía se limitaba a entregar lo que fue requerido

en el momento y no a necesidades posteriores como informes, indicadores y estadísticas respecto de la situación del centro de salud, o la obsolescencia de dicho servicio cuando la normativa de clasificación de pacientes cambiaba, lo que conllevó a contratar otro servicio que la aplique, contando así con múltiples proveedores parciales. Incluso si se tiene un proveedor del servicio de ficha clínica digital, aun son diversos proveedores y por lo tanto, formatos de ficha clínica con los que se trabaja, cada empresa tiene su estándar, lo que dificulta el seguimiento y transmisión de datos de un paciente si este se transfiere de un centro de salud a otro. Mejorar en estos aspectos es crucial, y es el ecosistema innovador quien actualmente está capacitando recursos humanos para responder a dichos requerimientos.

Hoy en día el aprendizaje de un lenguaje de programación es esencial para interactuar con la tecnología, como se mencionó previamente, se cuenta con la tecnología necesaria, pero el verdadero trabajo es adaptarla, transformarla para suplir las crecientes necesidades de digitalización, automatización de procesos y de información. Existe una gran cantidad de lenguajes de programación, así también los usos que se les puede dar, si se requiere crear una aplicación web, realizar análisis estadísticos, aplicar modelos matemáticos, analizar tendencias, diseñar un videojuego, utilizar sensores para obtener datos e incluso analizar la secuencia del ADN humano, para todas esas áreas un lenguaje de programación facilita gran parte de las tareas que se realizan manualmente en la actualidad, brindado tiempo a tareas que requieren de pensar más que repetir acciones.

Capítulo 5: Conclusiones

Transformar los datos provistos al formato Tidy Data permite descubrir el valor real de los datos, como los datos con los que se cuenta inicialmente no presentan una estructura estandarizada, ni datos homologados, la visualización final no será representativa del objeto de estudio, conteniendo métricas inconsistentes. El formato Tidy debe utilizarse para optimizar el tiempo de ejecución de las consultas a la base de datos y facilitar el manejo de las tablas para ser visualizadas.

Diseñar la base de datos relacional es una actividad fundamental para el desarrollo de plataformas de Inteligencia de Negocios, porque es la base con la cual se realizan las consultas a la base de datos, es un mapa que contiene las relaciones entre las tablas, el formato de cada una de sus columnas, y las tablas intermedias que permiten realizar consultas de alta complejidad, que finalmente generan conocimiento para la organización.

El Backend de una plataforma de Inteligencia de Negocios contempla las funciones o procedimientos (API) diseñadas para comunicar el Frontend con la base de datos, estas API tienen por objetivo estandarizar diversas aplicaciones de la plataforma permitiendo hacer uso de ellas sin estar dentro de la aplicación, por ejemplo, la API de Google Maps permite realizar consultas con la dirección de un lugar y retornar la longitud y la latitud de dicho lugar para ser utilizados en un modelo de ruteo si fuese el caso. Se debe tener en consideración el formato de la consulta para las API generadas puesto que estas son visibles para el usuario y pueden acceder en todo momento a ellas, y, por lo tanto, al contenido de la base de datos que esa API retorna, estos datos pueden ser sensibles como contraseñas o fichas clínicas lo que

hace necesario tener precauciones adicionales, la utilización de métodos POST o GET en este aspecto es esencial.

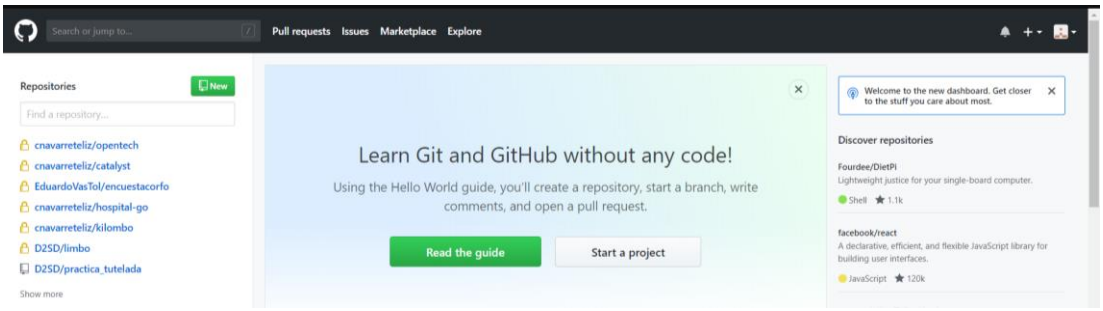
Implementar un prototipo de plataforma de Inteligencia de Negocios requiere de un cambio de paradigma respecto del valor que tienen los datos, como estos, dependiendo de su calidad y atomicidad (transacción más pequeña medible), pueden transformarse en información de la cual se obtiene nuevo conocimiento respecto del problema en estudio, para ello la gerencia de la institución que solicita una herramienta de BI debe estar dispuesta a realizar la inversión necesaria para utilizar los datos con los que se cuenta y adoptar protocolos diseñados para mejorar los datos que sean generados a futuro (estructura y calidad), no solo en el departamento donde se aplica la herramienta, sino que en todos aquellos que generan datos y no están siendo utilizados, porque en el auge de los datos todas las actividades son registradas, y por lo tanto todas pueden generar conocimiento para tomar mejores decisiones.

Referencias

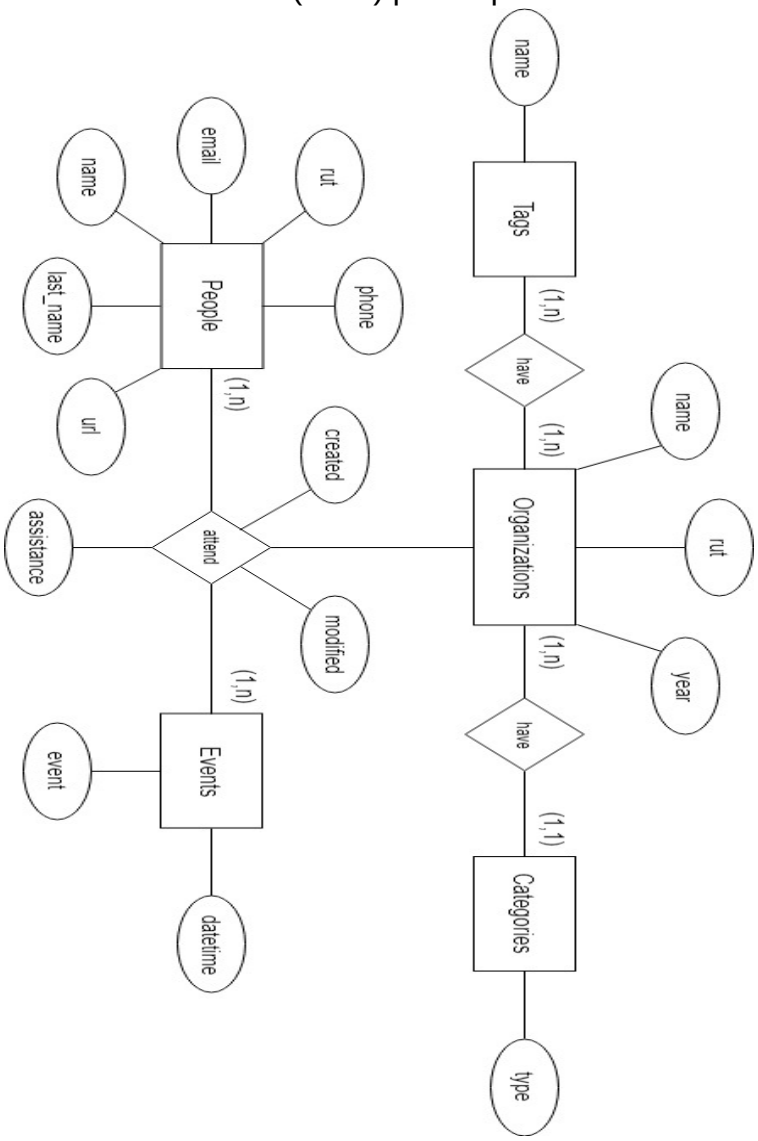
- Endel, F. Piringer, H.(2015).Data wrangling: Making data useful again. Enlace https://ac.els-cdn.com/S2405896315001986/1-s2.0-S2405896315001986-main.pdf?_tid=5e10c9a1-6bfa-4473-bbb9-3d5feaa3d701&acdnat=1546820432_486b2188de34bdfcaeae6f7cb67ec34c
- Gajare, P. Rangdale, S. (2015). ETL Data conversion: Extraction, Transformation and loading Data conversion. Enlace https://www.ijecs.in/index.php/ijecs/article/download/2574/2379/
- Holovaty, A. Willison, S, (2018). Django Documentation. Enlace https://media.readthedocs.org/pdf/django/2.1.x/django.pdf
- Kandel, S. Heer, J. Plaisant, C. Kennedy, J. Ham, F. Riche, N. Weaver, C. Lee, B. Brodbeck, D. Buono, P. (2011). Research directions in data wrangling: Visualizations and transformations for usable and credible data. http://vis.stanford.edu/files/2011-DataWrangling-IVJ.pdf
- Pedregosa et al. (2011). Scikit-learn: Machine Learning in Pyton. Enlace https://scikit-learn.org/stable/documentation.html
- Ronacher, A. (2018). Flask Documentation. Enlace http://flask.pocoo.org/docs/1.0/
- Wickham, H. (2014). Tidy Data. Enlace https://www.jstatsoft.org/article/view/v059i10

Anexos

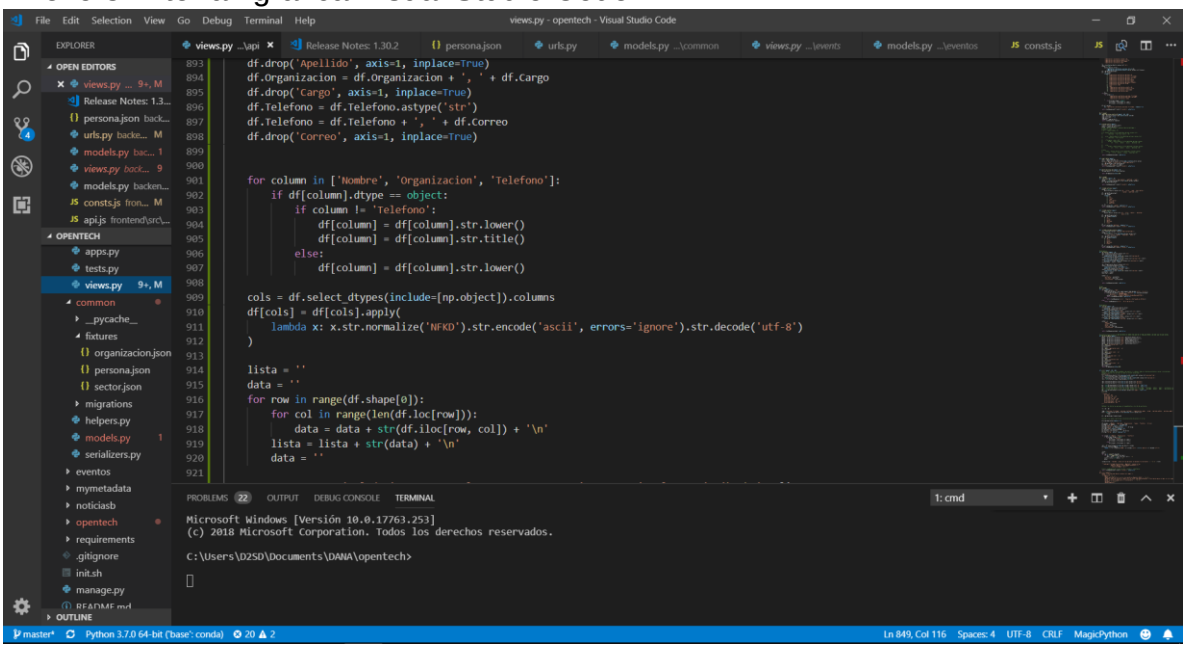
Anexo 1: plataforma de código colaborativo GitHub.



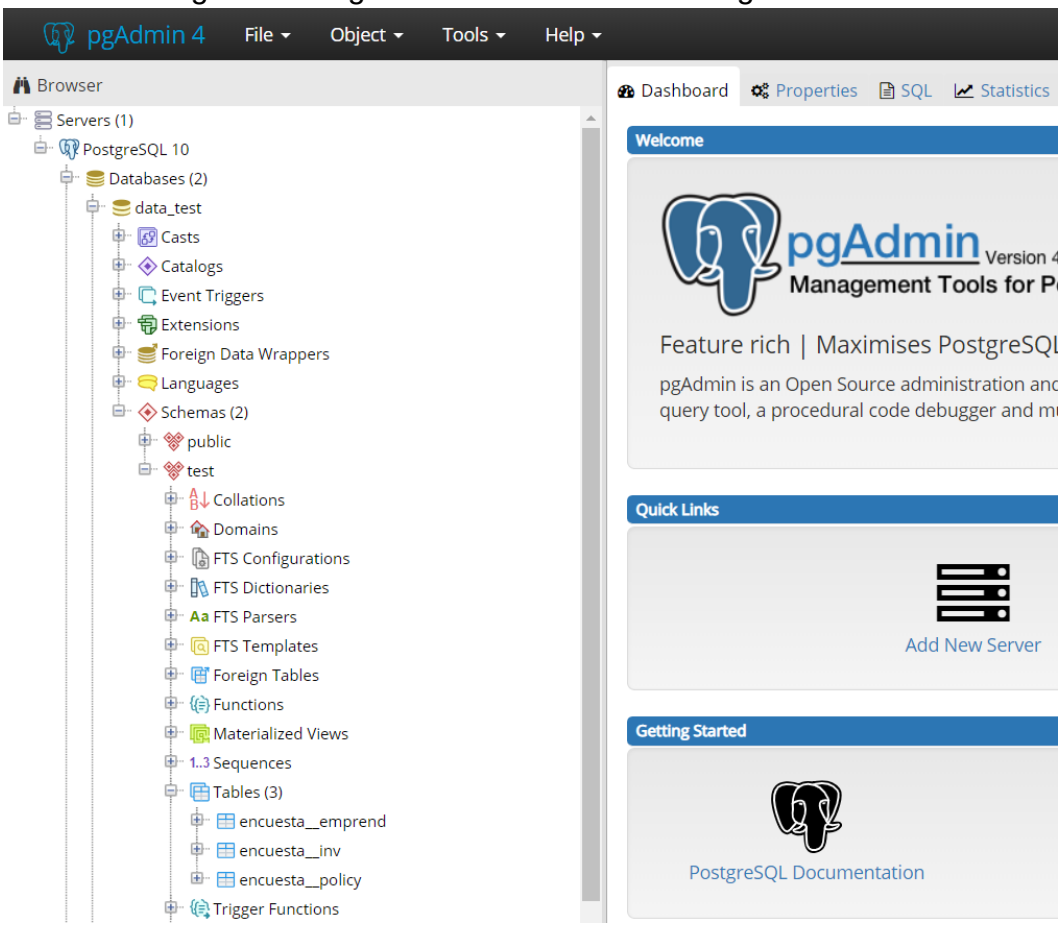
Anexo 2: Modelo Entidad-Relación (MER) para Open Tech.



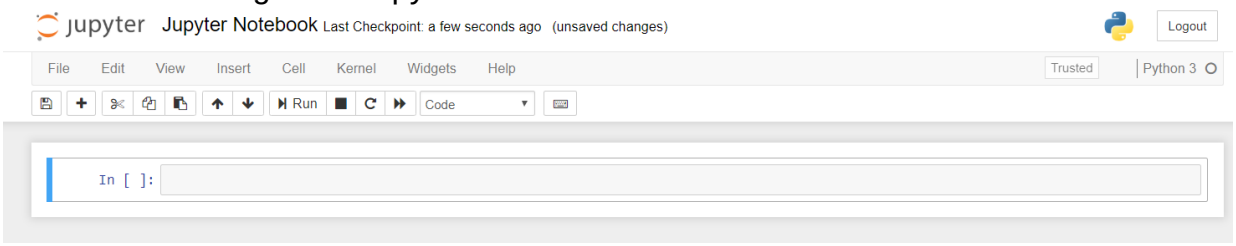
Anexo 3: interfaz gráfica Visual Studio Code.



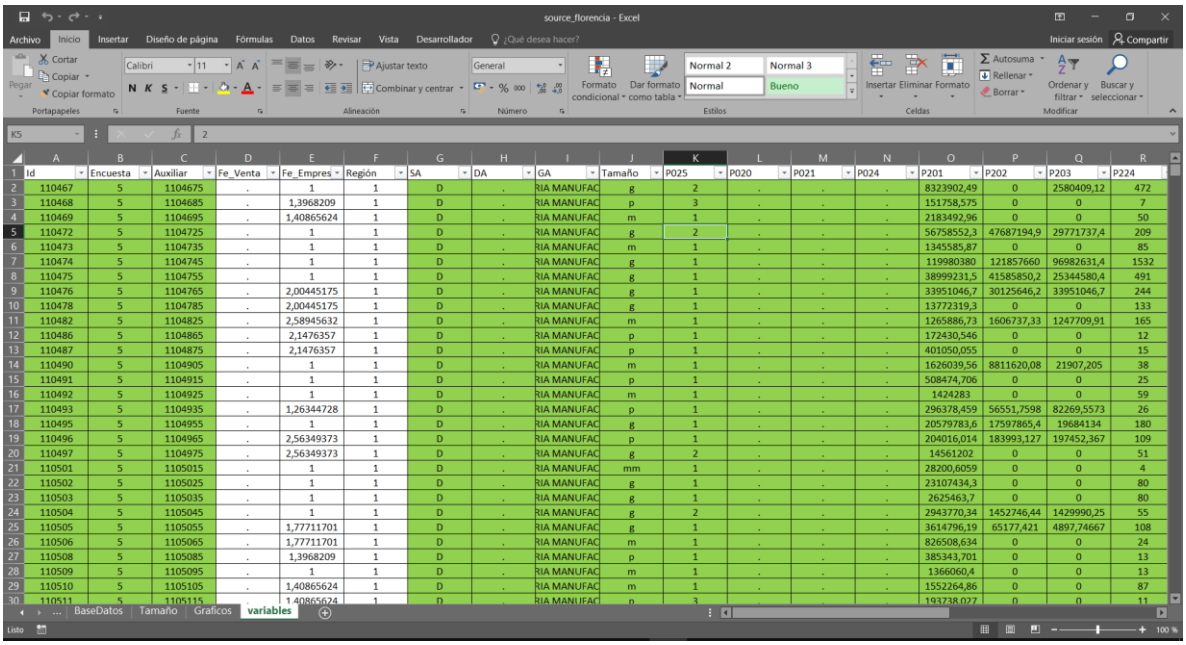
Anexo 4: interfaz gráfica del gestor de bases de datos PgAdmin



Anexo 5: interfaz gráfica Jupyter Notebook

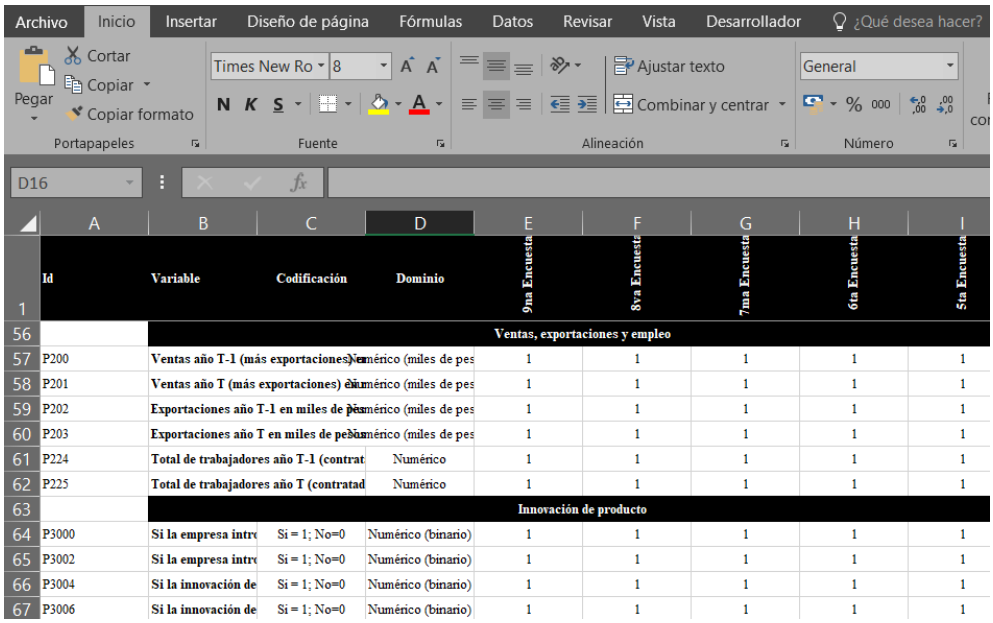


Anexo 6: base de datos Encuesta de Innovación.



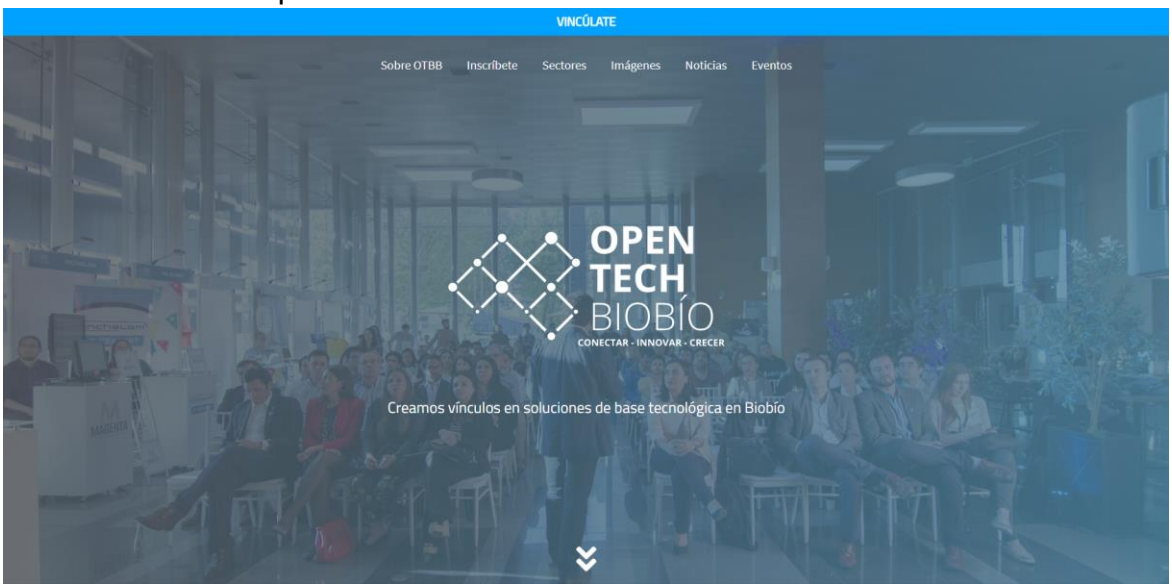
Id	Encuesta	Auxiliar	Fe_Venta	Fe_Empres	Región	SA	DA	GA	Tamaño	P025	P020	P021	P024	P201	P202	P203	P224
110467	5	1104675	.	1	1	D	.	RIA MANUFAC	g	2	.	.	.	8323902,49	0	2580409,12	472
110468	5	1104685	.	1,3968209	1	D	.	RIA MANUFAC	p	3	.	.	.	151758,575	0	0	7
110469	5	1104695	.	1,40865624	1	D	.	RIA MANUFAC	m	1	.	.	.	2183492,96	0	0	50
110472	5	1104725	.	1	1	D	.	RIA MANUFAC	g	2	.	.	.	56758552,3	47687194,9	29771737,4	209
110473	5	1104735	.	1	1	D	.	RIA MANUFAC	m	1	.	.	.	1345585,87	0	0	85
110474	5	1104745	.	1	1	D	.	RIA MANUFAC	g	1	.	.	.	119980380	121857660	96982631,4	1532
110475	5	1104755	.	1	1	D	.	RIA MANUFAC	g	1	.	.	.	38999231,5	41585850,2	25344580,4	491
110476	5	1104765	.	2,00445175	1	D	.	RIA MANUFAC	g	1	.	.	.	33951046,7	30125646,2	33951046,7	244
110478	5	1104785	.	2,00445175	1	D	.	RIA MANUFAC	g	1	.	.	.	13772319,3	0	0	133
110482	5	1104825	.	2,58945632	1	D	.	RIA MANUFAC	m	1	.	.	.	1265886,73	1606737,33	1247709,91	165
110486	5	1104865	.	2,1476357	1	D	.	RIA MANUFAC	p	1	.	.	.	172430,546	0	0	12
110487	5	1104875	.	2,1476357	1	D	.	RIA MANUFAC	p	1	.	.	.	401050,055	0	0	15
110490	5	1104905	.	1	1	D	.	RIA MANUFAC	m	1	.	.	.	1626039,56	8811620,08	21907,205	38
110491	5	1104915	.	1	1	D	.	RIA MANUFAC	p	1	.	.	.	508474,706	0	0	25
110492	5	1104925	.	1	1	D	.	RIA MANUFAC	m	1	.	.	.	1424283	0	0	59
110493	5	1104935	.	1,26344728	1	D	.	RIA MANUFAC	p	1	.	.	.	296378,459	56551,7598	82269,5573	26
110495	5	1104955	.	1	1	D	.	RIA MANUFAC	g	1	.	.	.	20579783,6	17597865,4	19684134	180
110496	5	1104965	.	2,56349373	1	D	.	RIA MANUFAC	p	1	.	.	.	204016,014	183993,127	197452,367	109
110497	5	1104975	.	2,56349373	1	D	.	RIA MANUFAC	g	2	.	.	.	14561202	0	0	51
110501	5	1105015	.	1	1	D	.	RIA MANUFAC	mm	1	.	.	.	28200,6059	0	0	4
110502	5	1105025	.	1	1	D	.	RIA MANUFAC	g	1	.	.	.	23107434,3	0	0	80
110503	5	1105035	.	1	1	D	.	RIA MANUFAC	g	1	.	.	.	2625463,7	0	0	80
110504	5	1105045	.	1	1	D	.	RIA MANUFAC	g	2	.	.	.	2943770,34	1452746,44	1429990,25	55
110505	5	1105055	.	1,77711701	1	D	.	RIA MANUFAC	g	1	.	.	.	3614796,19	65177,421	4897,74667	108
110506	5	1105065	.	1,77711701	1	D	.	RIA MANUFAC	m	1	.	.	.	826508,634	0	0	24
110508	5	1105085	.	1,3968209	1	D	.	RIA MANUFAC	p	1	.	.	.	385343,701	0	0	13
110509	5	1105095	.	1	1	D	.	RIA MANUFAC	m	1	.	.	.	1366060,4	0	0	13
110510	5	1105105	.	1,40865624	1	D	.	RIA MANUFAC	m	1	.	.	.	1552264,86	0	0	87
110511	5	1105115	.	1,40865624	1	D	.	RIA MANUFAC	p	3	.	.	.	193738,027	0	0	11

Anexo 7: codificación de columnas base de datos Encuesta de Innovación.



	Id	Variable	Codificación	Dominio	9na Encuesta	8va Encuesta	7ma Encuesta	6ta Encuesta	5ta Encuesta
56		Ventas, exportaciones y empleo							
57	P200	Ventas año T-1 (más exportaciones)		Numérico (miles de pes	1	1	1	1	1
58	P201	Ventas año T (más exportaciones)		Numérico (miles de pes	1	1	1	1	1
59	P202	Exportaciones año T-1 en miles de pes		Numérico (miles de pes	1	1	1	1	1
60	P203	Exportaciones año T en miles de pes		Numérico (miles de pes	1	1	1	1	1
61	P224	Total de trabajadores año T-1 (contrat		Numérico	1	1	1	1	1
62	P225	Total de trabajadores año T (contratad		Numérico	1	1	1	1	1
63		Innovación de producto							
64	P3000	Si la empresa intro	Si = 1; No=0	Numérico (binario)	1	1	1	1	1
65	P3002	Si la empresa intro	Si = 1; No=0	Numérico (binario)	1	1	1	1	1
66	P3004	Si la innovación de	Si = 1; No=0	Numérico (binario)	1	1	1	1	1
67	P3006	Si la innovación de	Si = 1; No=0	Numérico (binario)	1	1	1	1	1

Anexo 8: interfaz Open Tech Biobío.



Anexo 9: interfaz HOSPITAL GO.

samueloso21@gmail.com 07/05/19.