Facultad de Ingeniería

Escuela de Ingeniería en Bioinformática

Covid-19 en Chile y modelo epidemiológico SEIR

Taller de desarrollo de software 2020-2

Alumno:
Claudio Quevedo

Profesor:
Alejandro Valdés

Índice

1. Introducción

En este informe se presenta el desarrollo de un software en base al modelo epidemiológico matemático SEIR (susceptible, expuesto, infectado, recuperado) en las comunas de Chile. Esta software utilizara las bases de datos otorgadas por el Ministerio de Ciencia y Tecnología[1] junto a los datos entregados por el Ministerio de Salud de Chile[2], además de parámetros otorgados por el Colegio Médico de Chile[3].

Posterior a los cálculos realizados por el modelo, se aplicarán algoritmos de minería de datos e inteligencia artificial para la clasificación de las comunas en grupos de necesidad de recursos médicos (que pueden ser de extrema o baja necesidad). El propósito del software es la ayuda a la toma de decisiones para la distribución de implementos y de personal médico en las distintas localidades del país en base a su población y a las necesidades de éstas.

A su vez se hará uso de metodologías ágiles para su desarrollo. El desarrollo ágil de software se refiere a metodologías de desarrollo de software centradas en la idea de desarrollo iterativo, donde los requisitos y las soluciones evolucionan a través de la colaboración entre equipos multifuncionales auto-organizados. El valor máximo en el desarrollo ágil es que permite a los equipos entregar valor más rápido, con mayor calidad y predictibilidad, y con mayor aptitud para responder al cambio.

XP (eXtreme Programming) es una metodología ágil, fundada por Kent Beck, centrada en potenciar las relaciones interpersonales como clave para el éxito en desarrollo de software, promoviendo el trabajo en equipo, preocupándose por el aprendizaje de los desarrolladores, y propiciando un buen clima de trabajo.

2. Contexto general

2.1. Cliente

Chile también se a visto afectado por la pandemia mundial de covid-19, por lo cual el cliente objetivo es aquel que debe tomar las desiciones sobre como y donde se enviarán los recursos médicos disponibles en el país.

A su vez como será un servicio web, cumplirá un rol de servicio social para aquellas personas que quieran saber el estado de las predicciones en su propia comuna.

2.2. Problema o necesidad

El pasado 3 de Marzo de 2020 se confirmó el primer caso de Covid-19 en Chile, hoy 14 de Diciembre ya existen 573.830 casos confirmados[4]. Actualmente en Chile no hay ningún software que ayude a la toma de decisiones respecto a la distribución de recursos médicos basados en un análisis predictivo, en tiempo real (entregándole los datos al modelo) y personalizado para cada comuna del país.

El estudio de las enfermedades infecciosas a menudo se basa en modelos epidemiológicos matemáticos que intentan simular la dinámica de la enfermedad y estimar los parámetros relacionados con ella, como la tasa de reproducibilidad, velocidad de transmisión, fuerza de acción del gobierno, intensidad de reacción individual, entre otras.

Un tipo de simulación es el modelo SEIR basado en el supuesto de que la población se puede clasificar en cuatro grupos compartimentados independientes (individuos susceptibles, individuos expuestos, individuos infectados e individuos recuperados). Este modelo estudia de qué manera los individuos pueden progresar de un grupo compartimentado al siguiente.

2.3. Solución

Desarrollo de un software (servicio web) que utiliza el modelo SEIR para estudiar de qué manera los individuos pueden progresar de un grupo compartimentado al siguiente, esto ayudara a la toma de decisiones, tanto para estimar la necesidad de recursos médicos de regiones y comunas, como un servicio a la comunidad para conocer en que estado se encuentra su respectiva comuna.

El cual como input tiene el listado de cada comuna y su respectiva cantidad de habitantes, cantidad de infectados, cantidad de recuperados y su tasa de migración (o movimiento de personas entre comunas).

Luego la realización de los cálculos con el modelo SEIR y algoritmos de minería de datos e inteligencia artificial. Y finalmente el output será entregado en el servicio web, en el cual se puede ver en la Figura 1..

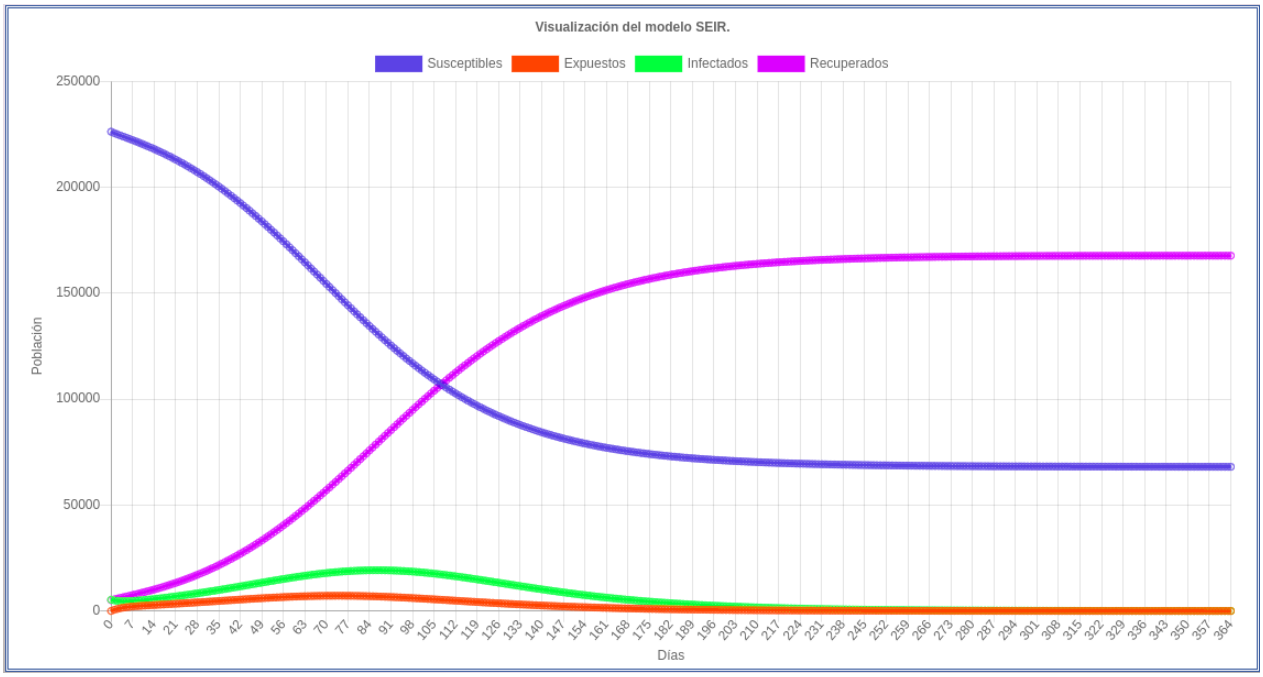

Figura 1: Ejemplo de gráfico del modelo epidemiológico matemático SEIR en una comuna. En el eje "y" se encuentra la población, en el eje "x" los días (un año). En la cual se observan las curvas lineales que se representan sin distanciamiento social.

3. Metodología

Se utilizará la metodología ágil XP (programación extrema), el cual es uno de los marcos de desarrollo de software más importantes[5]. Así lograr mejorar la calidad del software y responder a los requisitos del cliente. Además es una metodología que requiere de pocas personas (básicamente 2, pero yo actuaré de forma virtual como 2 personas).

XP se basa en la iteración frecuente a través de la cual los desarrolladores implementan historias de usuario. Las historias de usuario son declaraciones sencillas e informales del cliente sobre las funcionalidades necesarias[6].

- Una historia de usuario es una descripción convencional del usuario sobre una característica del sistema requerido.
- No menciona detalles más finos como los diferentes escenarios que pueden ocurrir. Sobre la base de historias de usuario, el equipo del proyecto propone "Metáforas". Las metáforas son una visión común de cómo funcionaría el sistema.
- El equipo de desarrollo puede decidir construir un "Spike" para alguna característica. Un Spike es un programa muy simple que se construye para explorar la idoneidad de una solución propuesta. Puede considerarse similar a un prototipo.

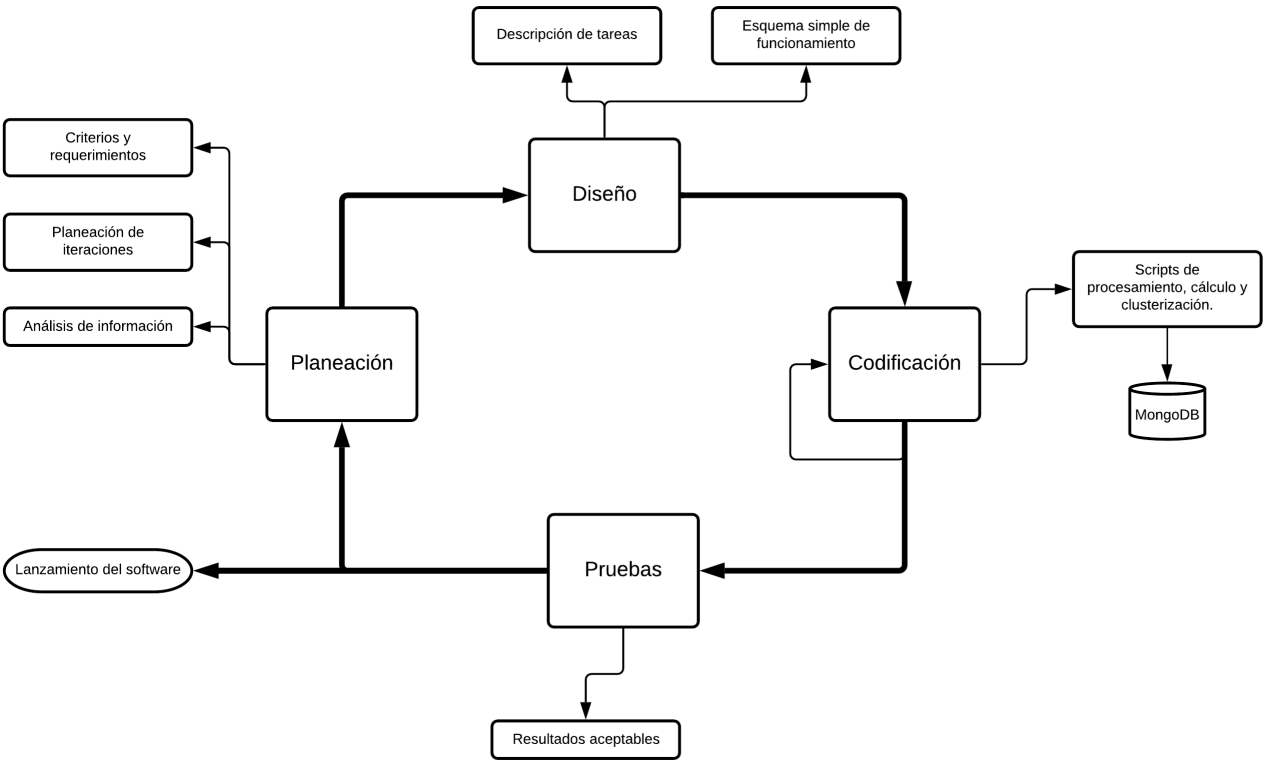

Figura 2: Representación de desarrollo del software utilizando la metodología XP.

En la Figura 2 se puede ver como primero se inicia con una planeación del proyecto, en el cual se debe hacer un análisis de la información, planear las iteraciones, anotar los criterios y requerimientos del sistema. Luego en el diseño se hará la descripción de tareas a desarrollar en dichas iteraciones utilizando un esquema simple de funcionamiento. Después de se realizara la codificación de los distintos scripts, ya sea de procesamiento de datos, de cálculo y de clusterización. Los cuales se guardarán en la base de datos no relacional MongoDB. Una vez terminada la codificación se realizarán pruebas, en las cuales se buscarán resultados relativamente aceptables para así poder lanzar el software a producción.

4. Desarrollo

4.1. Requerimientos

4.2. Diseño

El diseño general del sistema partirá con la utilización de los datos del Ministerio de Ciencia, que luego serán procesados para poder realizar los cálculos de las predicciones, luego esos resultados se almacenarán en una base de datos llamada MongoDB, después se ejecutarán los algoritmos de clusterización para hacer grupos de comunas y eso será mostrado en el servicio web.

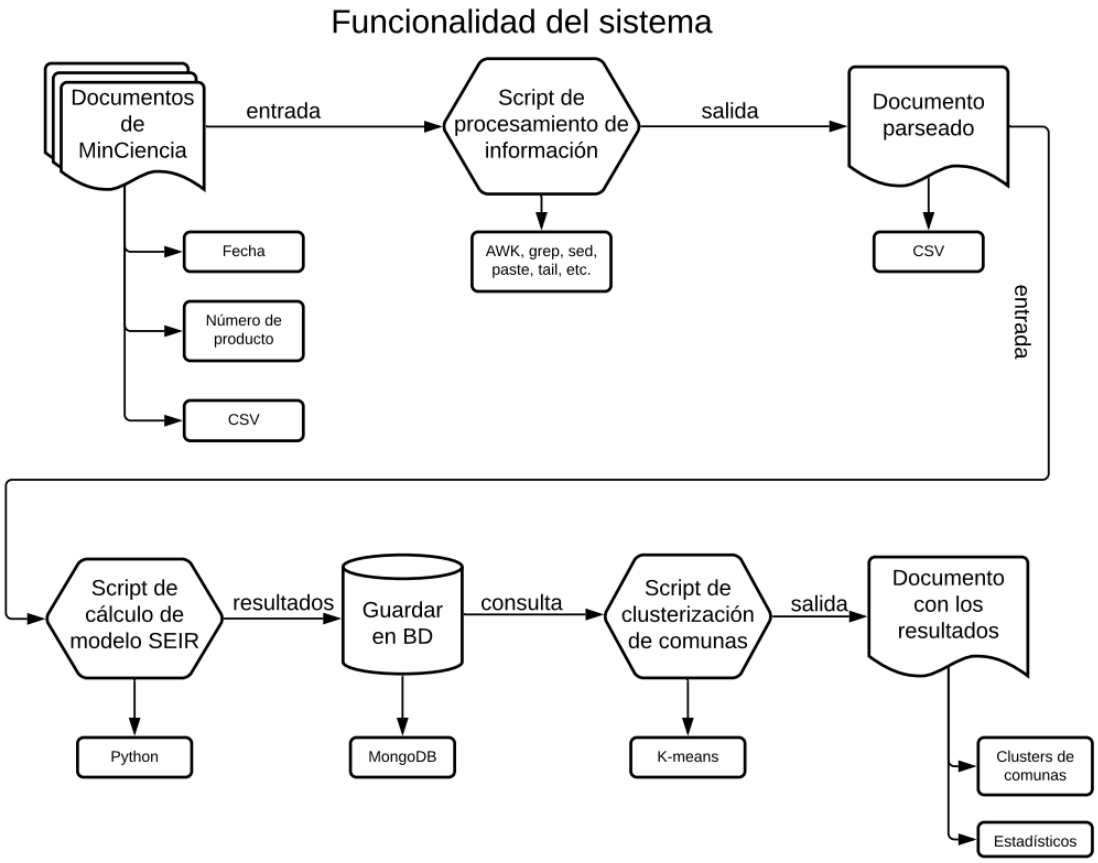

Figura 3: Funcionalidad general del sistema.

4.2.1. Base de datos

Se utilizará una base de datos no relacional llamada MongoDB, es una base de datos distribuida, basada en documentos y de uso general que ha sido diseñada para desarrolladores de aplicaciones modernas y para la era de la nube. Es una base de datos documental, lo que significa que almacena datos en forma de documentos tipo JSON[7].

Las razones del porqué no se utilizará una base de datos relacional como PostgreSql[8], es porque por un lado se quiere utilizar lo aprendido en el módulo anterior y por que el servicio web solo será una visualización de información, por lo tanto las consultas que se harán a la base de datos serán muy sencillas.

Se crearon 4 documentos: comuna (la cual tiene todos los resultados de la simulación SEIR), listacomuna (la cual tiene el listado de todas las comunas), listaregiones(la cual tiene el listado de todas las regiones) y estadísticas (la cual contiene toda la información estadística analizada).

Comuna (comuna):

- date = fecha en la que se realiza el cálculo del modelo.
- region = nombre de la región.
- comuna = nombre de la comuna.
- dia = día simulado con máxima cantidad de infectados.
- S = susceptibles.
- E = expuestos.
- I = infectados.
- R = recuperados.

Lista de comunas (listacomunas):

- region = nombre de la región.
- nombre = nombre de la comuna.

Lista de regiones (listaregiones):

- nombre = nombre de la región.

Resultados estadistícos (estadisticas):

- poblacion_porcentaje = porcentaje de la población infectada.
- poblacion = cantidad de habitantes de la comuna.
- comuna = nombre de la comuna.
- maxS = máximo de susceptibles.
- maxE = máximo de expuestos.
- maxI = máximo de infectados.
- maxR = máximo de recuperados.
- maxDia = día de la máxima cantidad de infectados.
- dia = fecha simulada con máxima cantidad de infectados.

4.2.2. Tecnologías

Las tecnologías a utilizar serán gratuitas y que tengan soporte al menos hasta 2021, las que cumplen con esto son las siguientes:

1. Git[9]: es un software de control de versiones que permitirá llevar un historial de cambios en los archivos del proyecto.
2. Gitlab[10]: servirá para crear un flujo de desarrollo de software optimizado y como repositorio del proyecto.
3. Bash[11]: encargado de leer y ejecutar órdenes desde un archivo.
4. Python 3.6[12]: es un lenguaje de programación interpretado, por lo que funciona en cualquier tipo de sistema que integre su interpretador. Las librerías utilizadas son las siguientes:
   - Djongo[13].
   - Pymongo[14].
   - Pandas[15].
   - Numpy[16].
   - Scipy[17].
   - Sklearn[18].
5. MongoDB[19]: es una base de datos de documentos con escalabilidad y flexibilidad que necesitaremos para desarrollar este software.
6. Django 3.1[20]: es un framework web de Python de alto nivel que fomenta un desarrollo rápido y un diseño limpio y pragmático. Se encarga de gran parte de la molestia del desarrollo web. Es gratis y de código abierto.
7. Bootstrap4[21]: es un kit de herramientas de código abierto para desarrollos web responsive (lo que permitirá verse bien en tablets y celulares) con HTML, CSS y JavaScript.
8. Chart.js[22]: para la creación de los gráficos JavaScript.
9. Fontawesome[23]: set de iconos que acompañaran a bootstrap4.
10. Jquery 3.2.1[24]: es una librería de JavaScript que facilita el desarrollo de páginas web dinámicas e interactivas. Ofrece muchas funciones para simplificar el desarrollo del front-end de un sitio, incluyendo la capacidad de realizar solicitudes asíncronas.

4.3. Implementación

A continuación se detallan cada una de las iteraciones planificadas durante el transcurso del desarrollo del software, a medida que una iteración se finalizaba, se continuaba con la siguiente, exceptuando la iteración paralela, que se ejecutó durante todo el proyecto.

4.3.1. Iteración 1, descarga y procesamiento de información.

1. Revisar y descargar la información entregada por el Ministerio de Ciencia, Tecnología, Conocimiento e Innovación (disponible en https://github.com/MinCiencia/Datos-COVID19).
2. En la cual se filtrarán desde el producto 2 los casos confirmados de cada comuna.
3. Desde el producto 38 para filtrar los fallecidos por comuna.
4. No hay información de recuperados, así que se tendrá que calcular por fecha de inicio de los síntomas.
5. Luego de descargar los datos de casos confirmados, fallecidos y el documento con las fechas de inicio y término de síntomas, se procede a no descargar toda la información de todas las fechas (debido a su baja o nula consistencia), por lo cual se tomará la información más actualizada (solo lo de las últimas 2 semanas).
6. Se extraerá información de la migración de cada comuna desde el censo 2017[25].
7. Programar código que hará el procesamiento de los datos para que quede un archivo csv con las siguientes columnas: nombre comuna, región a la que pertenece la comuna, población, contagiados, recuperados, fallecidos, tasa de migración entre comunas.
8. Se aplaza 1 semana mas, debido a problemas técnicos.

4.3.2. Iteración 2, programación de scripts para calcular el modelo SEIR y guardar sus resultados.

1. Este script se codificará con el lenguaje de programación Python (versión 3.6) y la utilización de las librerías Numpy, Scipy y Mongo.
2. Como input será un archivo en formato csv, con las columnas nombre comuna, región a la que pertenece la comuna, población, contagiados, recuperados, fallecidos, tasa de migración entre comunas.
3. Teniendo la información, se realizará el cálculo del modelo SEIR.

4. Guardar esta información resultante post-cálculo en la base de datos no relacional MongoDB.

5. Se aplaza 1 semana, por problemas técnicos.

6. Atraso, se aplaza 3 días mas.

4.3.3. Iteración 3, clusterización de comunas y generar documento con los resultados.

1. Normalizar los datos utilizando el escalado estándar.

2. Dentro del script de python, generar documento con los resultados obtenidos.

3. Aplicar algoritmos de minería de datos e inteligencia artificial.

4. Para realizar esta clusterización se utilizarán datos resultantes del modelo SEIR, los cuales serán el día en el que se prediga el máximo de contagios, el número de contagios, los expuestos ese día y los recuperados ese día.

5. Para la clusterización se utilizará el algoritmo K-mean.

6. Clusterización de comunas con mayor y menor necesidad de recursos médicos en Chile.

7. Se aplaza 2 días, debido a que se avanzo primero con Django y mongoDB.

4.3.4. Iteración 4, visualización, testeos y pruebas de funcionalidad del software.

1. Se programará página web para mostrar resultados.

2. Se revisará la calidad de los datos que se le entregan al script por si es necesario hacer algún cambio dentro de los parámetros.

3. Esto se hará analizando los resultados obtenidos, si tienen coherencia o no (por ejemplo, que agrupara una comuna con muy pocos casos de contagio con comunas altamente contagiadas, donde claramente dicha comuna no necesitaría prioridad en instrumentación médica).

4. Revisar utilización de herramientas gráficas para mostrar resultados.

5. Usuario podrá seleccionar su comuna y ver lo resultados de esa comuna sin necesidad de cargar todos los dados de todas las comunas de Chile.

6. Aprender a utilizar método POST y GET en django. Crear página principal. Crear página de las comunas (pero no una página por cada comuna, tiene que ser dinámica).

7. Crear documentos en MongoDB.
8. Crear documento mongoDB y visualización Django de Regiones.
9. Visualizar comunas en base a su estado (Extrema, alta, medio y baja necesidad).
10. Agregar más información en los resultados estadísticos. Indicar día en que empezarán a haber más recuperados que susceptibles. Indicar el día además de la fecha. Indicar el % de la población infectada.
11. Realizar testing y pruebas del software en general.
12. Actualizar informe.

4.3.5. Iteración paralela, análisis de creación del proyecto y visualización de la data.

1. Testear Django y como funciona con la base de datos escogida.
2. Creación del proyecto en Django.
3. Actualizar informe, traspasar lo hecho a LaTeX, mejorar aspectos, etc.
4. Si bien se escogió MongoDB, ver la posibilidad de usar base de datos relacionales, todo esto en base a que seá lo que el software debe mostrar.
5. Análisis de la base de dato, que sistema será conveniente, que es lo que se guardará, en base a lo que se mostrará de los resultados pre calculados.

4.3.6. Mejoras del software en su versión V2.0

Se anotaron problemas actuales del software, para luego arreglar o mejorar. Estas mejoras están consideradas para el año 2021.

1. Mejorar el procesamiento de los datos. Actualmente como el software lo está haciendo en base a la población que tiene la comuna, hay unos errores respecto a lo que se ve en los noticiarios. Por ejemplo, Temuco queda en el cluster de las comunas que tienen una baja necesidad de recursos médicos, cuando es una de las comunas con más contagios, este error es debido a que la comuna de Temuco tiene una población grande respecto a otras comunas. El posible arreglo es no considerar la población de la comuna o plantearla de una forma distinta en el datasets.

2. Mejorar la cantidad de clusters. Actualmente están: Extremo, alto, medio y bajo. Pero tal vez deberían ser mas clusters, quedaría pendiente este análisis.

3. Integrar vacunación al modelo matemático.

5. Conclusiones

Se logró desarrollar la primera versión del software que simula el modelo epidemiologico SEIR de cada comuna de Chile. Se hizo una prediccion del estado en que se encuentra cada comuna tomando en cuenta la población, obteniendo el dia en el que la simulacion indica una mayor cantidad de infectados, y a partir de este dia analizar la cantidad de susceptibles, expuestos y recuperados. Luego tomar este conjunto de informacion y aplicar el algoritmo de clusterizacion para determinar en que estado esta la comuna.

Por otro lado, para desarrollar el proyecto se utilizó la metodologia ágil XP, junto a esta metodología se hizo uso de la herramienta Gitlab para mantener un control en la realización del proyecto. Si bien en un inicio fue complicado adecuarse a esta metodología debido a la falta de experiencia, se pudo sacar adelante durante este semestre, semestre especial en medio de la pandemia. Una vez que se hizo costumbre el uso de gitlab y de las iteraciones, la organización del desarrollo del software fue mucho mas fácil. Además en el caso de que el cliente hiciera alguna consulta, estaría todo en en la plataforma (lo cual ayudaría bastante en caso de tener problemas).

Sin embargo hubo otras complicaciones aparte de adecuarse a la metodología, estas complicaciones principalmente surgieron por los cambios constantes en la arquitectura de la entrega de los datos por parte del Ministerio de Ciencias. Otro problema que tomo bastante tiempo lograr resolver es la visualización de los resultados, en primera instancia se iba a crear un documento con los resultados de todas las comunas y sus estados, pero luego se decidió en que era mejor para el usuario un servicio web. Lo anterior es debido a que en primer lugar el usuario debía descargar los archivos del Ministerio de Ciencias, y en base a los archivos que se descargaran, el software entregaría los resultados. Esto último no resultaba óptimo para el usuario, lo cual sería un problema serio, puesto que nadie querría usar una herramienta que es difícil de utilizar (que sí era, puesto que no es fácil buscar un archivo csv en medio de mas de 60 productos que esta liberando el Ministerio de Ciencias en su repositorio).

Finalmente, si bien la curva de aprendizaje de Django es bastante tosca, una vez que se aprende es bastante más rápido que utilizar la forma lenta de ir programando absolutamente todo. Anteriormente había intentado aprender Django pero no resultó, ahora gracias a un tutorial entregado por el profesor, la curva de aprendizaje fue menos grotezca.

Referencias

[1] Ministerio de Ciencia
https://www.minciencia.gob.cl

[2] Ministerio de Salud
https://www.minsal.cl

[3] Colegio Médico
http://www.colegiomedico.cl

[4] Casos Confirmados
https://www.minsal.cl/nuevo-coronavirus-2019-ncov/casos-confirmados-en-chile-covid-19/

[5] XP
https://www.geeksforgeeks.org/software-engineering-extreme-programming-xp/

[6] XP 2
http://www.jtech.ua.es/j2ee/2011-2012/restringido/met/sesion02-apuntes.html

[7] MongoDB
https://www.mongodb.com/es

[8] PostGresQL
https://www.postgresql.org

[9] GIT
https://git-scm.com

[10] GitLab
https://gitlab.com

[11] BASH
https://www.gnu.org/software/bash/

[12] Python 3.6
https://docs.python.org/3/whatsnew/3.6.html

[13] Djongo
https://www.djongomapper.com

[14] Pymongo
https://pymongo.readthedocs.io/en/stable/

[15] Pandas
https://pandas.pydata.org

[16] Numpy
https://numpy.org

[17] Scipy
https://www.scipy.org

[18] Sklearn
https://scikit-learn.org/stable/

[19] MongoDB
https://www.mongodb.com/es

[20] Django
https://docs.djangoproject.com/en/3.1/releases/3.1/

[21] Bootstrap4
https://getbootstrap.com/docs/4.0/getting-started/introduction/

[22] Chart.js
https://www.chartjs.org

[23] Fontawesome
https://fontawesome.com

[24] Jquery
https://jquery.com

[25] CENSO2017
https://www.censo2017.cl