Máster en Bioinformática y Bioestadística - UOC / UB Febrero 2022 - Semestre de primavera

**Trabajo de Final de Máster - PEC 1: Plan de Trabajo**

---

**Alumno**: Juan Vigueras Díaz

**Tutor**: Jorge Valencia Delgadillo

**Título alternativo propuesto**: Comparación de modelos de inteligencia artificial basados en estructuras y descriptores moleculares para predicción de bioactividad en el marco del desarrollo de fármacos contra las bacterias *Escherichia coli*, *Staphylococcus aureus* y *Pseudomonas aeruginosa*.

---

**Contexto y Justificación del Trabajo**: El desarrollo de fármacos constituye una serie de procesos de larga durabilidad en los que además se invierte una gran cantidad de recursos económicos. Ser capaces de detectar de manera precoz la inviabilidad de una molécula como posible fármaco implicaría el ahorro de estos recursos en favor de enfocar los estudios hacia propuestas con un potencial de éxito mayor. No obstante, múltiples factores entran en juego en el complejo sistema que representa la interacción entre el patógeno y un agente externo (molécula), por ejemplo, los distintos sistemas de defensa que pueden presentar las bacterias en su pared celular o el análisis y entendimiento del mecanismo de acción que desencadena la actividad de la molécula sobre la diana terapéutica. Los algoritmos de inteligencia artificial, que en términos generales se pueden entender como modelos capaces de realizar una abstracción matemática de las relaciones intrínsecas entre las variables de entrada y las de salida, pueden resultar muy útiles en este marco de trabajo. Actuando como modelos de caja negra, se podría conseguir una predicción de bioactividad sin la necesidad de analizar y considerar previamente las características bioquímicas del escenario de estudio, para poder finalmente así realizar una primera valoración de la viabilidad farmacológica de una molécula.

**Descripción general**: Bajo el contexto anteriormente expuesto, el trabajo se enfoca en la comparación de varios tipos de algoritmos de inteligencia artificial (*deep learning, machine learning*) para la predicción del nivel de bioactividad en tres bacterias comúnmente conocidas por ser las causantes de diversas patologías: *E. coli, S. aureus* y *P. aeruginosa*; así como la evaluación de viabilidad de esta tecnología como herramienta complementaria en la etapa inicial de investigación pre-clínica de una o varias moléculas candidatas. Los modelos de Deep Learning tendrán como entrada los grafos correspondientes a las estructuras de dichas moléculas, mientras que los modelos de Machine Learning tendrán como entrada un conjunto de descriptores moleculares calculados a partir de las mismas. Con esto, se pretende estudiar la diferencia de rendimiento entre utilizar un conjunto de variables calculadas a partir de la estructura molecular y utilizar directamente la estructura en sí misma. Se utilizarán los datos relativos a tres especies de bacterias con tal de probar el nivel de escalabilidad de esta herramienta sobre distintos organismos y para no limitar el estudio a un único ejemplar, en cuyo caso no se sabría si los resultados vendrían intrínsecamente dados por haber escogido esa especie en concreto. Además, dada la disponibilidad de los datos y el carácter del estudio en el que, como ya se ha mencionado, no se contempla la bioquímica de los mecanismos de acción, se seleccionará una enzima de cada bacteria para estudiar tanto la bioactividad a nivel proteico como la bioactividad patógena (a nivel bacteriano). Por último, como indicadores de bioactividad se usarán el IC50 (mitad de la concentración inhibitoria máxima) tanto para bacterias como para enzimas y el MIC (concentración mínima inhibitoria) únicamente para bacterias. Estas medidas, obtenidas en experimentos en los que se realizan múltiples mediciones, resultan más fiables que otros indicadores como, por ejemplo, la *actividad*, que se obtiene mediante una única medición.

**Objetivos**: Siguiendo lo anteriormente expuesto conjuntamente con la línea de objetivos descritos en la PEC anterior y ajustándolos a las pequeñas modificaciones introducidas hasta ahora, los objetivos de este trabajo se agrupan de la siguiente manera:

**Objetivos generales**:

1. Determinar la viabilidad, en términos de rendimiento de modelo, de aplicación de algoritmos de inteligencia artificial en la predicción de bioactividad.

2. Comparar, en dicha tarea de regresión, el uso entre variables de tipo estructural y variables de tipo descriptor molecular, contrastando los rendimientos obtenidos en cada caso.

3. Comparar el rendimiento de los modelos aplicados sobre una enzima concreta para cada una de las bacterias (bioactividad enzimática) y sobre las bacterias al completo (bioactividad patógena).

**Objetivos específicos**: Los objetivos generales anteriormente descritos se pueden subdividir y desarrollar en:

1. Calcular descriptores moleculares y grafos a partir de moléculas en formato SMILES obtenidas de una base de datos de bioactividad.

2. Definir y construir una arquitectura de Deep Learning desde cero capaz de procesar grafos en una tarea de regresión.

3. Entrenar los modelos de Deep Learning y Machine Learning implementados para la tarea de regresión y ajustar los hiper-parámetros de cada modelo en sucesivos experimentos para llegar al valor óptimo en las métricas de evaluación utilizadas (todavía por determinar; probablemente siendo la raíz del error cuadrado medio o RMSE).

4. Analizar el resultado de los experimentos para:

4. 1. Determinar el tipo de modelo que presenta mejor resultado en las métricas de evaluación - basado en estructuras o basado en descriptores moleculares.

4. 2. En caso de tratarse de un modelo Deep Learning, determinar la medida de bioactividad (IC50 o MIC) que presenta mejor resultado en las métricas de evaluación.

4. 3. Determinar contra qué bacteria y/o contra qué enzima se obtienen mejores resultados en las métricas de evaluación utilizadas o si, por contraparte, se obtienen resultados similares.

4. 4. Con esto, inferir la viabilidad del uso de algoritmos de inteligencia artificial en las etapas iniciales del proceso de desarrollo de nuevos fármacos.

5. Por último, en el caso de obtener como mínimo un modelo con resultados aceptables para ser aplicados en un contexto real, diseñar e implementar una aplicación que proporcione una predicción de bioactividad dada una molécula en formato SMILES.

**Enfoque y método a seguir**: La metodología *CRISP-DM* (CRoss Industry Standard Process for Data Mining) es un conocido método de trabajo comúnmente aplicado en proyectos industriales de ciencia de datos y que consta de las siguientes seis fases:

1. Entendimiento de negocio.
2. Entendimiento de los datos.
3. Preparación de datos.
4. Modelado.
5. Evaluación. Retorno a punto 1 en caso de no obtener resultados deseados.
6. Puesta en producción.

En este contexto, se puede modificar ligeramente esta metodología para conseguir un método adaptado a un proyecto de inteligencia artificial aplicado en el ámbito académico:

1. Planteamiento del problema. Corresponde al contexto y justificación del trabajo.

2. Entendimiento de los datos. Corresponde a determinar si la base de datos con la que se trabajará proporciona la información necesaria para el correcto desarrollo del proyecto.

3. Preparación de los datos. Corresponde al objetivo específico [1.].

4. Modelado. Corresponde al objetivo específico [2.] y a la parte del objetivo [3.] relativa a cada uno de los entrenamientos por separado.

5. Evaluación. Corresponde a la parte del objetivo [3.] relativa a la evaluación del entrenamiento realizado y a la vuelta al paso anterior de manera cíclica para refinar los modelos en sucesivos entrenamientos; y al objetivo específico [4.] una vez se ha hecho el suficiente número de experimentos.

6. Una vez realizada la inferencia y hechas las conclusiones pertinentes, implementación de la aplicación con el modelo seleccionado(s) y creación de un repositorio con el código desarrollado y los resultados obtenidos hasta llegar a la solución final.

Esta metodología aplicada para la tarea de predicción que se desea estudiar supone el estado del arte en este ámbito. Para desarrollar el procedimiento expuesto, se programará un conjunto de *scripts*, trabajando en un entorno de máquina local. Al no disponer de unidad de procesamiento gráfico (GPU), el tratamiento de grafos en el caso de los modelos de Deep Learning podría resultar en tiempos de cálculo relativamente largos, con lo que se considerará utilizar la herramienta 'Google Collab' (que dispone de una GPU gratuita de hasta 15 GB de capacidad dependiendo del servidor asociado) si de forma local los entrenamientos resultan demasiado lentos. El lenguaje de programación escogido será Python 3.0 dada la disponibilidad de los *frameworks* de desarrollo gratuitos Tensor Flow, Pytorch y Scikit-Learn para trabajar con Deep Learning y Machine Learning (así como otras librerías relacionadas con el procesamiento de datos y otras tareas a realizar).

**Planificación**:

**Tareas**: A continuación se enumeran las tareas que corresponden al cumplimiento de los objetivos de este trabajo. La duración de cada una de las actividades se detalla en el diagrama de Gantt del siguiente apartado.

1. Descarga de datos: Descarga, desde la base de bioactividad 'ChEMBL', de los datos relativos a las bacterias de estudio y sus correspondientes enzimas (una para cada organismo). La información se recupera desde la sección 'Targets', que incluye información sobre las moléculas testeadas.

| Tabla a descargar | | Campos esenciales* a descargar de cada tabla | | |
|---|---|---|---|---|
| Nombre (Target) | Código en la BBDD | Cadena SMILES | IC50 | MIC |
| *Escherichia coli* | CHEMBL354 | X | X | X |
| *Dihydrofolate reductase* | CHEMBL1809 | X | X | |
| *Staphylococcus aureus* | CHEMBL352 | X | X | X |
| *DNA gyrase subunit B* | CHEMBL3038482 | X | X | |
| *Pseudomonas aeruginosa* | CHEMBL348 | X | X | X |
| *Beta Lactamase* | CHEMBL1293246 | X | X | |

* Queda por determinar si se escogen campos adicionales relativos a información (descriptores) sobre las moléculas, dado que estos campos serán igualmente calculados en tareas posteriores, aunque podrían ser de utilidad para comprobar que el cálculo se realiza de forma correcta.

En un principio, la enzima se escoge teniendo en cuenta aquella que más moléculas testeadas tiene sobre la medida IC50. No se distingue ni se contempla si la enzima pertenece al exterior de la bacteria (pared celular) o si se encuentra en el interior de esta, así como las implicaciones farmacológicas que esto supondría. A lo largo del trabajo, estas enzimas podrán ser sustituidas por otras que, pese a tener menos registros, presenten una distribución de valores de bioactividad con mayor rango dinámico de aplicación. Este análisis está pendiente de realizar en la primera fase del proyecto y cualquier modificación al respecto se indicará en la siguiente PEC.

Los datos se descargan de forma local en formato de fichero 'CSV', uno para cada tabla.

2. Cálculo de indicadores moleculares y grafos: Con la librería RdKit de Python y a partir de la cadena de caracteres en formato SMILES de cada una de las moléculas, cálculo de descriptores moleculares (todavía por determinar, tanto la cantidad como los descriptores seleccionados; la elección final se indicará en la siguiente PEC) y de la tabla química (MOL File) correspondiente. A partir de la tabla química (que incluye las coordenadas de átomos y los tipos de enlace) y con la librería Pytorch Geometric, cálculo de los grafos correspondientes a las moléculas en los que las aristas están categorizadas según el tipo de enlace químico y los vértices (o nodos) están categorizados según el tipo de átomo.

Los indicadores serán el *input* de los modelos de Machine Learning, y el posible pre-procesamiento de datos adicional (normalización de datos, codificación de variables, etc.) se indicará en la siguiente PEC. Los grafos serán el *input* del modelo de Deep Learning.

3. Implementación de modelos ML y DL:

Para la tarea de regresión a partir de grafos (estructuras) y con la librería Pytorch de Python, implementación de un modelo de Deep Learning conformado por tres capas convolucionales, las dos primeras seguidas de una unidad de normalización y una unidad ReLU y la última seguida de una unidad *max pooling*, y una capa lineal (perceptrón) cuyo *output* sea el valor numérico de la medida de bioactividad correspondiente (IC50 o MIC). La técnica de Dropout, de ser necesaria, se aplicaría en la tercera capa convolucional. De disponer del tiempo suficiente o de presentar problemas de sobre-ajuste, se considerará implementar y evaluar la misma arquitectura únicamente con una y/o dos capas convolucionales (lo cual se indicará en las siguientes PEC).

Para la tarea de regresión a partir de indicadores moleculares y con la librería Scikit-Learn de Python, importación de los siguientes modelos de Machine Learning: X-Gradient Boost, Random Forest y Support Vector Machine. Estos algoritmos vienen directamente implementados en la librería y se invocan simplemente creando un objeto mediante la instancia de la clase correspondiente.

Los hiper-parámetros iniciales con los que se implementarán cada uno de los modelos se detallarán en la siguiente PEC.

4. Implementación de los *loops* de 'train' y 'test':

Implementación de un bucle de entrenamiento de modelos. Para el modelo de Deep Learning, un ciclo que incluya el método de retropropagación y la evaluación sobre el conjunto de datos de validación que servirá para seleccionar el mejor modelo a partir del mejor resultado obtenido para la métrica de evaluación en cada una de las épocas (ciclos). Para los modelos de Machine Learning, el ajuste viene automáticamente implementado en el método correspondiente.

Implementación de un bucle de evaluación de modelos. Para el modelo de Deep Learning, un ciclo similar al de validación. Para los modelos de Machine Learning, la evaluación viene automáticamente implementada en el método correspondiente

Del mismo modo que la implementación de los modelos, estos bucles se programarán con las librerías Pytorch y Scikit-Learn de Python.

5. Prueba de entrenamiento: En primer lugar, realización de una prueba *feed forward* para corroborar que los grafos recorren correctamente el algoritmo de Deep Learning implementado y se obtiene un *output* sin errores. Esta prueba es imposible de hacer para los modelos de Machine Learning, que ya vienen directamente implementados y listos para el entrenamiento y es en este punto donde, de haberlos, se observarían los errores. A continuación, ejecución de un entrenamiento de prueba para todos los modelos para comprobar que todos los algoritmos disponen de capacidad de aprendizaje. Este factor se evaluará observando las curvas de pérdida para los modelos de Deep Learning y las curvas de aprendizaje para los modelos de Machine Learning.

6. Implementación de automatización de experimentos: A partir de los bucles de la tarea número [4.], construcción de una función de entrenamiento completo de modelos que permita la ejecución directa de un experimento a partir del modelo y el conjunto de hiper-parámetros indicados.

7. Entrenamientos sucesivos y refinamiento de los modelos: A partir de la función implementada en el paso anterior, realización de entrenamientos sucesivos para refinar los modelos a partir de la modificación de sus hiper-parámetros o de la introducción (únicamente en el caso de modelos de Deep Learning) de métodos de regularización (como *Dropout*) o de prevención de *overshoot* (como *Scheduler*). Se irán guardando los *checkpoints* de cada entrenamiento. Se utilizará la herramienta TensorBoard de la librería TensorFlow para hacer un seguimiento de los resultados de los modelos de Deep Learning a partir de las curvas de pérdida, mientras que para hacer un seguimiento de los resultados de los modelos de Machine Learning a partir de las curvas de aprendizaje se utilizará la librería Scikit-Learn.

8. Implementación de repositorio GitHub y App: Creación de un repositorio en la plataforma GitHub con el código desarrollado en el proyecto. Se mostrarán:

- Los resultados obtenidos en los experimentos y los correspondientes *checkpoints*
- El código para utilizar un modelo para evaluar una o varias moléculas
- El código para ejecutar un experimento
- El conjunto de datos necesarios
- De desarrollarse, el código correspondiente a la aplicación

9. Redacción de PECs y memoria: Redacción de las dos PECS de seguimiento del trabajo (Fases 1 y 2 que se muestran a continuación en el Calendario) y de la memoria escrita del proyecto (correspondiente a la entrega de la PEC4).

10. Elaboración de presentación: Desarrollo de las *slides* de la presentación del proyecto, recogiendo los resultados más relevantes obtenidos en los experimentos así como el planteamiento inicial, la solución propuesta, los objetivos cumplidos y los hitos conseguidos, entre otros apartados. Grabación de un vídeo de presentación del proyecto siguiendo las directrices indicadas en la guía.

**Calendario**: Al final de este documento se muestra un calendario con las tareas del plan de trabajo anteriormente desarrolladas, que en algunos casos se agrupan o unifican. El calendario está basado en un diagrama de Gantt, y se ha realizado con la herramienta gratuita y online 'Monday'.

**Hitos**: Los hitos de este proyecto están directamente relacionados con las tareas a realizar:

1. Construir grafos y calcular descriptores moleculares a partir de moléculas en formato SMILES.
2. Diseñar e implementar un modelo de Deep Learning desde cero.
3. Entrenar los modelos de Deep Learning y Machine Learning con los datos obtenidos en el paso 1.
4. Conseguir una mejora sucesiva de los modelos en los experimentos realizados tras el ajuste de híper-parámetros. Como el resultado alcanzable en un proyecto de inteligencia artificial no se puede saber *a priori*, no se establece ningún valor como hito.
5. Analizar e inferir conclusiones a partir de los resultados obtenidos.
6. Desarrollo e implementación de la aplicación.

**Análisis de riesgos**: Se pueden dividir los análisis de riesgos en factores de alcanzabilidad (relacionados con las características específicas de este proyecto) y factores temporales:

Factores de alcanzabilidad:

- Dada la naturaleza de los algoritmos de inteligencia artificial, nada asegura que vaya a existir un resultado positivo (i.e., un modelo útil y aplicable) antes de realizar el desarrollo del estudio.
- Las bacterias o enzimas escogidas pueden no resultar útiles en el estudio por diversos motivos. Por ejemplo, falta de variabilidad en los indicadores de bioactividad. En este caso queda abierta la opción de reconsiderar la elección de otra bacteria o enzima, en cuyo caso se detallaría en la posterior PEC.
- El hecho de trabajar, en el caso de modelos de Deep Learning, con estructuras bidimensionales sin tener en cuenta el mecanismo de unión entre la molécula y la diana terapéutica marca un ejercicio sin precedentes encontrados en la bibliografía que podría resultar en la obtención nula de resultados,
- En el improbable pero posible caso en el que ninguno de los modelos estudiados resulte lo suficientemente apto, no tendría sentido el desarrollo de la aplicación para la simulación de lo que podría ser su puesta en producción.

Factores temporales:

- Puede existir insuficiencia de tiempo para realizar el número de experimentos suficientes como para optimizar los modelos.
- Las tareas están ajustadas a un calendario que deja cierto margen de actuación en caso de aparecer imprevistos, errores en código, resultados muy inesperados en los que se necesite profundizar, etc.

Aun así, estos factores pueden alterar el transcurso y/o realización de las tareas en cuanto a su distribución temporal y ejecución.

**Resultados esperados**:

Experimentales:

A nivel experimental, no se espera *a priori* un resultado positivo (alta relación entre las variables de entrada y las de salida) o negativo (ninguna relación en absoluto) ya que determinar el tipo de resultado es uno de los objetivos del proyecto. Además, dada la naturaleza del problema, así como en tareas de clasificación se podría establecer un nivel de métrica de evaluación deseada o esperada (por ejemplo, obtener una precisión entre 0.75 y 0.95), en este caso caso no es evidente establecer un valor para la métrica de evaluación, dado que el rango de esta depende del rango dinámico de las variables de salida y, en este escenario, estas variables no tienen un intervalo de valores acotado.

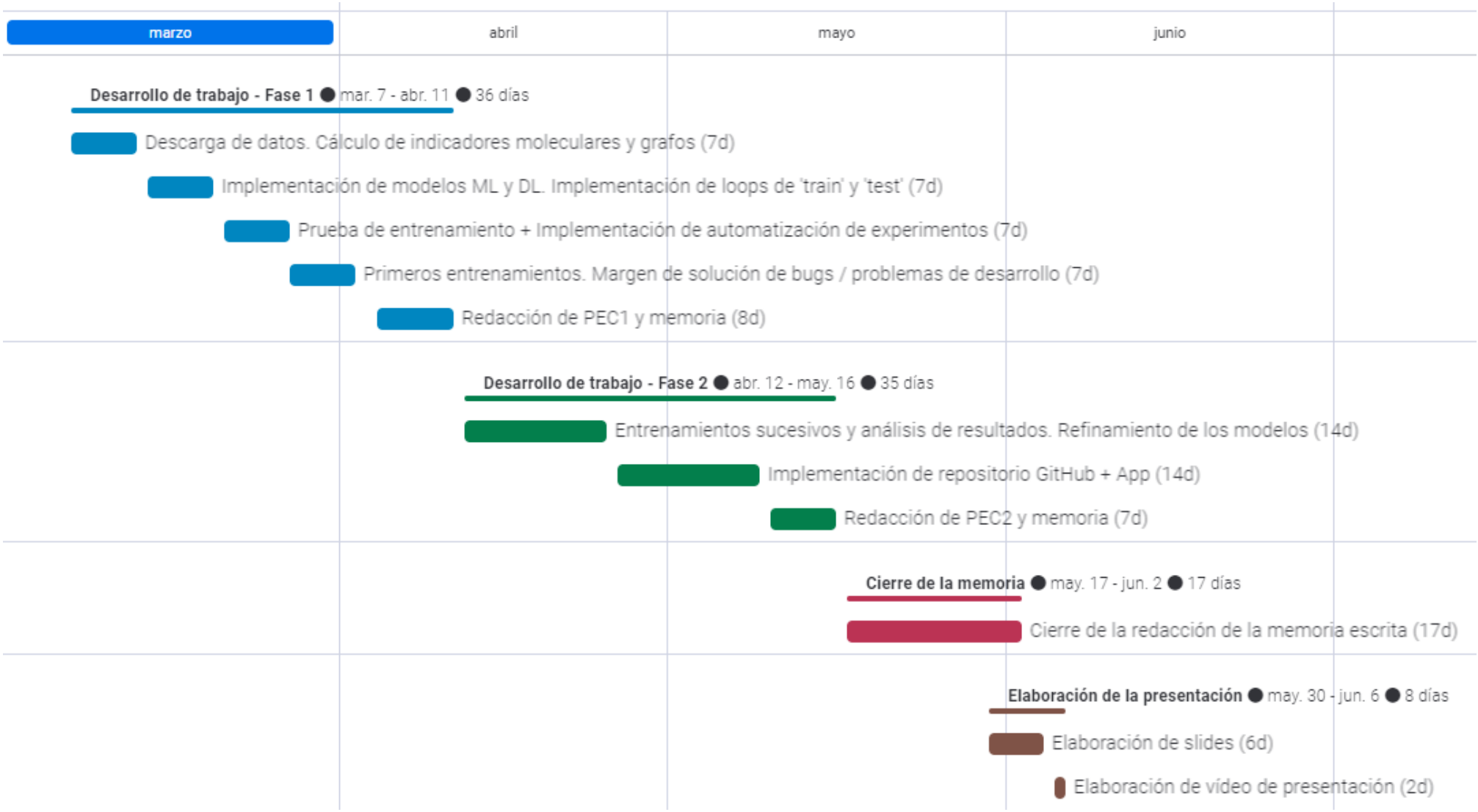
Entregables:

1. Plan de trabajo: Incluyen las PEC 0 (ya entregada) y 1 (presente documento) en las que se detallan tanto la propuesta de trabajo como el plan de desarrollo del mismo.

2. Memoria: Incluye, por una parte, las PEC 2 y 3 de estado de avance del proyecto en cada una de sus dos fases principales, así como la PEC 4 correspondiente a la memoria final del trabajo que recoge toda la información relativa al proyecto, siguiendo las directrices de la guía proporcionada.

3. Producto: Repositorio de GitHub con el código desarrollado en el trabajo, así como los experimentos realizados y los resultados obtenidos. En caso de obtener un resultado aplicable en un entorno real, desarrollo de una aplicación que permita obtener una predicción de bioactividad en base a una molécula dada. Las características de esta aplicación se determinarán en futuras PEC en función de los resultados obtenidos en los experimentos realizados.

4. Presentación virtual: Grabación en vídeo de la exposición del proyecto realizado que incluye las *slides* de soporte del mismo a partir de las directrices indicadas en la guía proporcionada.


marzo
abril
mayo
junio
Desarrollo de trabajo - Fase 1 ● mar. 7 - abr. 11 ● 36 días
Descarga de datos. Cálculo de indicadores moleculares y grafos (7d)
Implementación de modelos ML y DL. Implementación de loops de 'train' y 'test' (7d)
Prueba de entrenamiento + Implementación de automatización de experimentos (7d)
Primeros entrenamientos. Margen de solución de bugs / problemas de desarrollo (7d)
Redacción de PEC1 y memoria (8d)
Desarrollo de trabajo - Fase 2 ● abr. 12 - may. 16 ● 35 días
Entrenamientos sucesivos y análisis de resultados. Refinamiento de los modelos (14d)
Implementación de repositorio GitHub + App (14d)
Redacción de PEC2 y memoria (7d)
Cierre de la memoria ● may. 17 - jun. 2 ● 17 días
Cierre de la redacción de la memoria escrita (17d)
Elaboración de la presentación ● may. 30 - jun. 6 ● 8 días
Elaboración de slides (6d)
Elaboración de vídeo de presentación (2d)