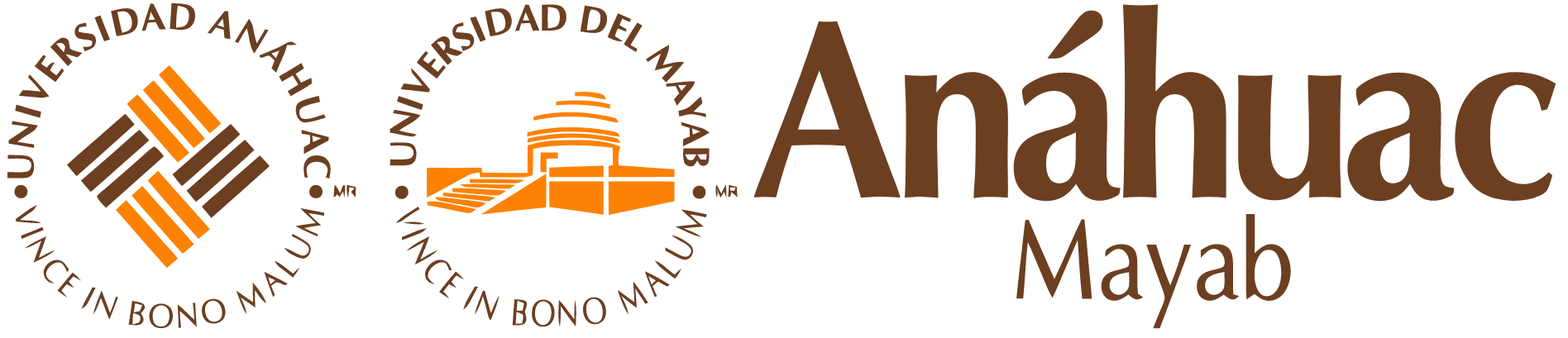
****

**Universidad Anáhuac del Mayab**

Lenguajes Orientados a Objetos

Prof. José Luis Barrera Canto

**“Proyecto Final: Damas Inglesas”**

****

Elaborado por:

Daniel Alejandro Rodríguez Arceo

Eiralhí Escobedo Marrufo

Carlos Bermejo Colli

Fecha de entrega: 13/06/2020

**Índice**

[**Descripción** 1](#_Toc45542967)

[**Especificaciones de desarrollo** 2](#_Toc45542968)

[**Clases** 3](#_Toc45542969)

[**Eventos** 4](#_Toc45542970)

[**Reflexión** 5](#_Toc45542971)

[**Conclusiones** 6](#_Toc45542972)

# **Descripción**

El proyecto consiste en el desarrollo de unas damas inglesas utilizando dos lenguajes de programación, uno que permite mostrar una interfaz gráfica de usuario (GUI) y otro en versión de consola.

Las damas es un juego diseñado para dos contrincantes que disponen de 12 piezas, llamadas peones, de un mismo color (pueden ser negras o blancas) para jugar y moverse en un tablero de 64 casillas. El objetivo del juego es capturar todas las fichas del oponente o inmovilizarlas de manera que no puedan moverse sin ser capturadas. El jugador que tiene las fichas negras comenzará la partida. Un peón sólo puede desplazarse hacia adelante una casilla a la vez (no puede moverse ni hacia atrás, ni a la derecha, ni a la izquierda) y en diagonal; los peones avanzan a una casilla contigua cuando está vacía o en diagonal saltando a una ficha oponente para comerla; se pueden realizar varias capturas en un mismo turno, siempre y cuando el que captura esté en una casilla adyacente al capturado, y que la casilla inmediatamente detrás de éste esté libre para que acabe el movimiento. Cabe recalcar que la captura es obligatoria. Si un jugador consigue llevar una de sus fichas al lado contrario del tablero, se convierte en una dama o reina, lo que significa que además de poder moverse en diagonal, también puede ir hacia delante y hacia atrás. La partida finaliza cuando un jugador se queda sin fichas o cuando tiene los movimientos bloqueados.

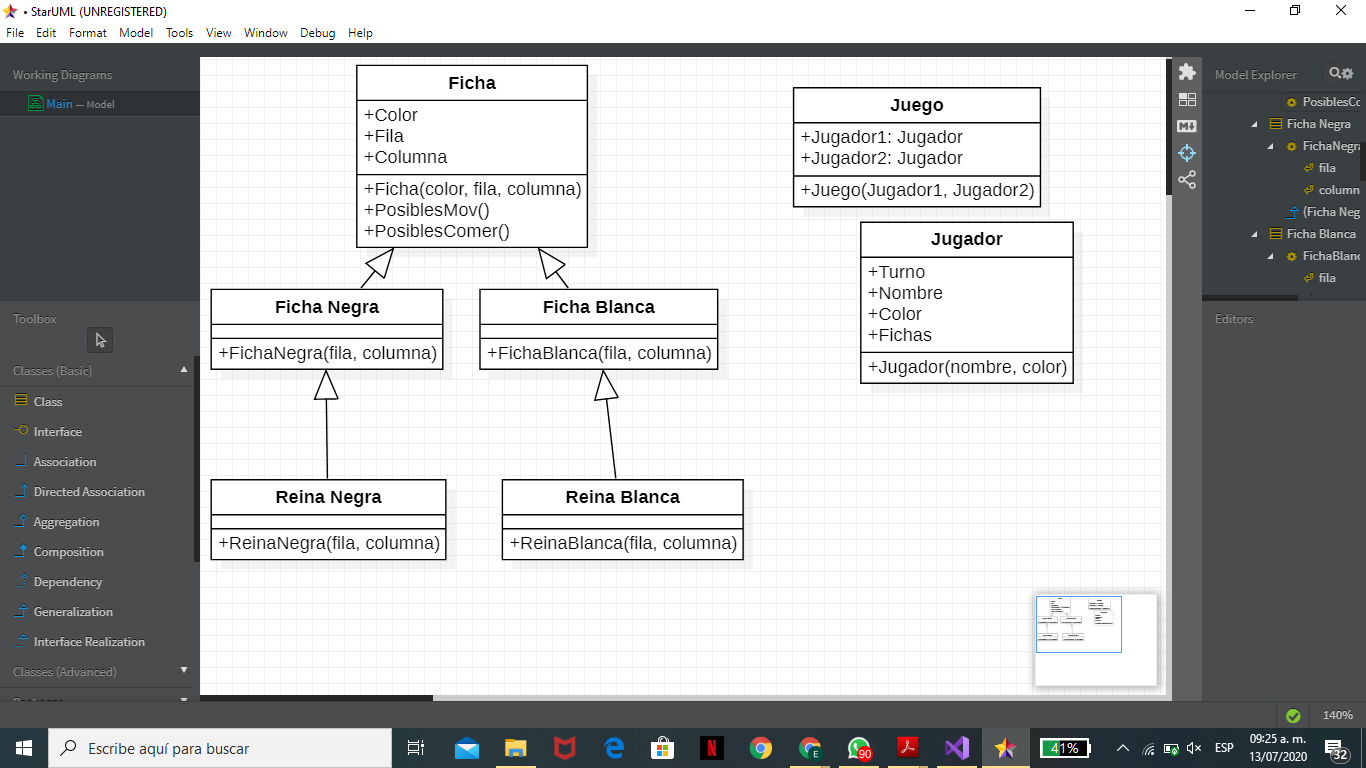
El propósito de nuestro proyecto es permitir que las personas interactúen y se diviertan. Jugar damas inglesas, al ser un juego de estrategia, trae diversos beneficios, como por ejemplo: ayuda a mejorar la concentración y la creatividad, potencia la memoria, estimula la capacidad de aprendizaje, incrementa el pensamiento crítico y favorece a la sana convivencia.

# **Especificaciones de desarrollo**

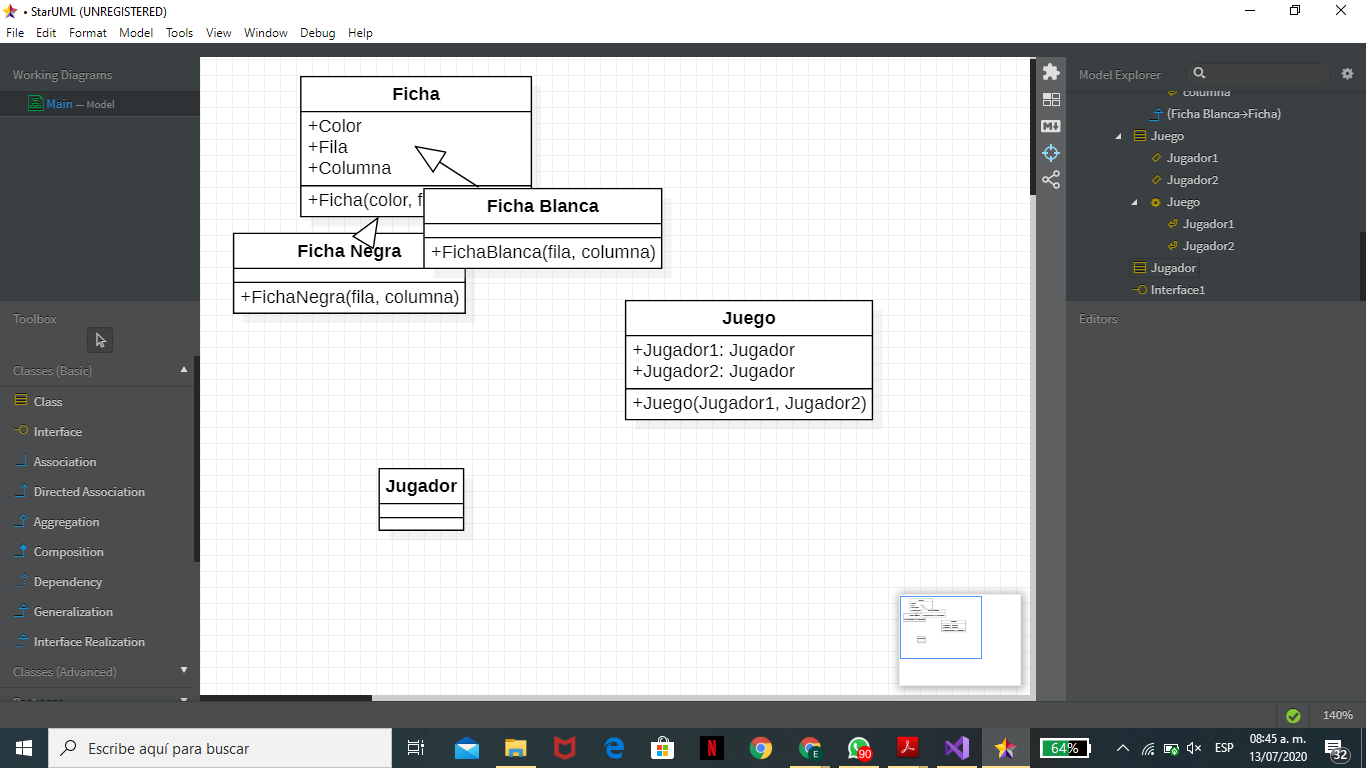
Para la versión del juego con interfaz gráfica de usuario usamos el entorno de desarrollo Visual Studio y el lenguaje de programación C#. Una de las razones por las cuales decidimos usar lo anterior, es la experiencia previa que se tenía en dicho lenguaje, por lo que resultaba más cómodo desarrollar el juego en C#. Sin embargo, ese no es el único motivo; a continuación, se mencionan algunas de las ventajas que tiene el lenguaje: se pueden declarar propiedades dentro de cualquier clase para así poder usar las clases como objetos; permite que los atributos de acceso puedan ser de tipo público, protegido, interno, interno protegido y privado; C Sharp tiene un rango muy amplio y definido de tipos de datos. Se puede agrupar o aislar código mediante namespaces.El lenguaje tiene una sintaxis sencilla que permite que sea más fácil escribir código y comprenderlo, sin embargo, debido a lo completo y específico que es, el proceso se vuelve más laborioso.

Para la versión en consola se usó Python en el entorno de desarrollo Spyder, este lenguaje reduce el tiempo que lleva escribir código, pues la sintaxis es bastante simple. De igual manera, es muy flexible, ya que ni siquiera es necesario declarar el tipo de datos. A pesar de ser un lenguaje simplificado, Python cuenta con una estructura ordenada y limpia que permite tener un código legible y comprensible para cualquier programador. Un inconveniente que pudimos notar de dicho lenguaje es que muchas veces no te marca los errores que tiene tu programa, por lo que te dejará correrlo, sin embargo, se comportará de manera diferente a la esperada. De igual forma, al tener una sintaxis diferente a los otros lenguajes que vimos durante el curso, resulta difícil comenzar a programar con Python, sin embargo, una vez que lo comprendes, se vuelve el más simple de escribir y comprender.

# **Clases**

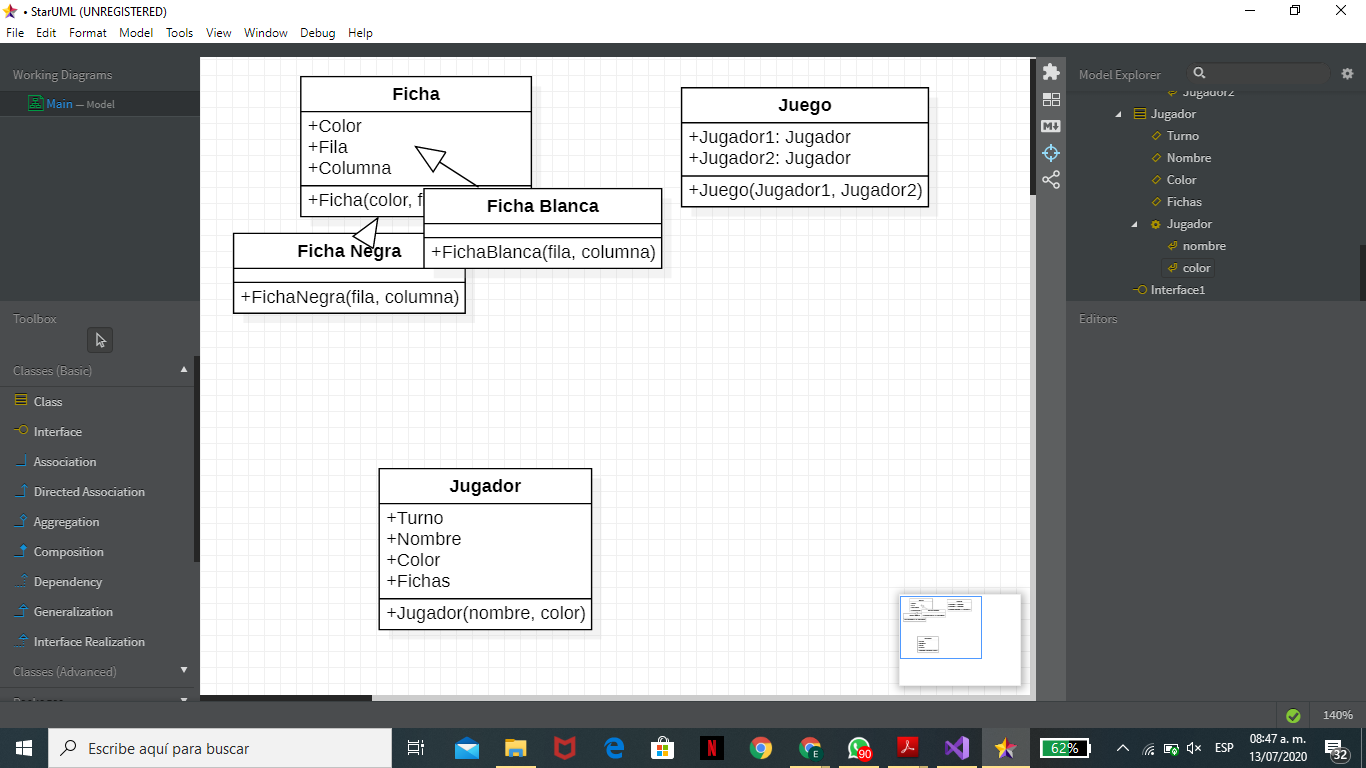
****

Se usaron estas clases para representar a las piezas del juego. Primero se tiene a la clase **Ficha** que tiene las propiedades color, fila y columna, su constructor y los métodos PosiblesMov() y PosiblesComer() que tienen la función de determinar cómo se pueden mover las piezas en el tablero y a qué espacios pueden desplazarse para capturar, respectivamente. Dicho comportamiento cambia según el tipo de ficha que sea. **Ficha Negra** y **Ficha Blanca** heredan de **Ficha; Reina Negra** hereda de **Ficha Negra**; **Reina Blanca** hereda de **Ficha Blanca**.

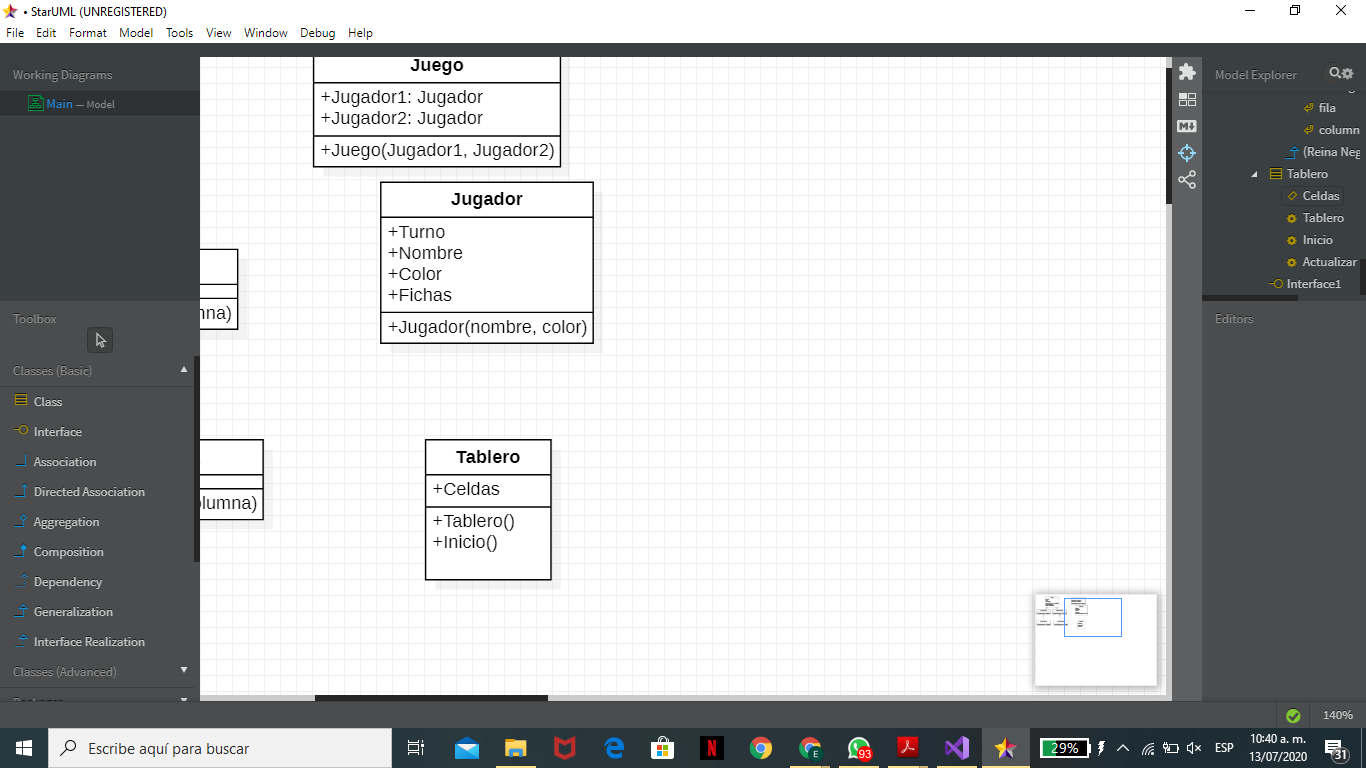
****

La clase **Juego** tiene dos propiedades que posteriormente son recibidas como parámetros de su constructor.

Su función es almacenar información de los dos jugadores, para así, facilitar su manipulación en el main.

****

La clase **Jugador** tiene cuatro propiedades y usa nombre y color como parámetros de su constructor. Sirve para representar al usuario que participará en el juego.

****

La clase **Tablero** representa las posiciones que ocupan las fichas blancas y las fichas negras. El método Inicio() se encarga de asignar las posiciones iniciales de las piezas.

# **Eventos**

Los eventos son acciones, como teclear o hacer click, que sirven para que el objeto reconozca que se está interactuando con él. Cuando un evento ocurre, el programa comienza a ejecutar el código correspondiente a dicho evento.

En el programa que se desarrolló implementamos el evento MouseClick, el cual ejecuta una serie de instrucciones cuando el usuario presiona el mouse.

# **Reflexión**

**Daniel Rodríguez:** En sí lo más complejo de desarrollar en el proyecto fue la lógica de los movimientos y la implementación de las distintas validaciones que se tienen que hacer en los distintos escenarios que pueden surgir a lo largo de una partida. Comenzamos desarrollando el proyecto de una manera, pero nos dimos cuenta de que nos estábamos alejando del paradigma de objetos y que nuestras clases no mostraban una relación fuerte entre sí por lo que tuvimos que rehacer el proyecto desde 0 para poder llegar a un mejor resultado.

**Eiralhi Escobedo:** El primer obstáculo que tuvimos se presentó durante el proceso de abstracción, ya que las clases que planteamos al principio no eran óptimas, por lo que tuvimos que volver a hacer el proyecto para que de esa forma, el programa funcionara bien. Durante el proceso de escritura del código, un problema significativo fue el no poder hacer que las piezas se movieran de la forma esperada y que se hicieran las validaciones necesarias para que el usuario pueda interactuar con el juego siguiendo las reglas de las damas inglesas.

**Carlos Bermejo:** El obstáculo más grande que encontramos, a mi parecer, fue la lógica de los movimientos de las piezas, en especial cuando el jugador escogía capturar las piezas de su oponente, algunos errores en el código o sintaxis los descubrimos cuando ya creíamos haber finalizado el desarrollo. La representación del tablero en Python fue todo un reto, ya que procuramos que, al igual que en la interfaz gráfica, el usuario vea en consola una representación amigable de las damas inglesas.

# **Conclusiones**

**Daniel Rodríguez:** En lo personal me gustó mucho el curso, antes de la primera clase sabía muy poco del concepto de orientación a objetos, sabía que existían las clases y objetos pero nunca los había puesto en práctica. Me di cuenta de lo útil que es esta materia y la importancia que tiene en diversos ámbitos especialmente si te enfocas en el desarrollo de software. En lo personal he estado trabajando en algunos proyectos con conocidos externos de la Mayab o incluso personas de otros estados y muchas veces hacíamos cosas simples o al menos en mi caso era así porque no había llevado esta materia antes, sin embargo, ahora me siento más confiado en cuanto a la cantidad de cosas que puedo hacer como programador y en cuanto a la calidad de mis proyectos. La materia fue bastante cansada y en momentos estresante, sin embargo, estoy contento de haberla tomado con el profesor Barrera ya que siento que a pesar de la dificultad que impone en sus clases, es opacada por el conocimiento que uno obtiene de ellas.

**Eiralhi Escobedo:**  Siento que todo el proceso del desarrollo del proyecto fue bastante complejo y nos permitió poner en práctica e implementar las habilidades y herramientas adquiridas durante el curso. Tuvimos que pensar muy bien cómo íbamos a realizar el programa, desde la etapa de abstracción en la que necesitamos identificar qué elementos necesitamos para elaborar el juego, hasta la parte del código en la que la lógica nos fallaba por momentos. En general, siento que termino el curso con muchos aprendizajes nuevos, pues antes de la materia, ni siquiera entendía qué era el paradigma de objetos, sin embargo, aún necesito seguir practicando y leyendo, pues se me dificulta entender y reconocer por completo todo lo que necesito a la hora de desarrollar un programa. El proyecto y la materia me motivaron para tomar cursos extracurriculares y seguir aprendiendo sobre los distintos lenguajes de programación.

**Carlos Bermejo:** Creo que ha sido más que un desafío. Para mí, poderme adaptar al ritmo de esta asignatura ha sido una labor especialmente difícil, mis nociones de programación eran muy básicas y puso a prueba mi capacidad para adaptarme (y pasar noches sin dormir).

Creo que la elaboración de este proyecto ha sido una montaña rusa, justo cuando uno creía haber dominado el programa, uno descubre hay muchos aspectos que no ha considerado para el adecuado funcionamiento del mismo.  Sin embargo, considero que he avanzado mucho, a pesar de las dificultades he logrado desarrollar un gusto por esta asignatura y la programación, espero en el futuro poder conservar este placer y continuar desarrollándome en el área.