**Estrategia de solución**

Para llevar a cabo la realización del proyecto de la consultora informática, plantearemos nuestra estrategia de solución en este documento, la cual organizaremos en diferentes categorías, que serán: Precondiciones, Clases, y Operaciones. Cada una de ellas será explicada detalladamente a continuación.

**Precondiciones:**

Las precondiciones que asumiremos al realizar este proyecto serán las siguientes:

* Solo existirán 3 niveles para los desarrolladores(I o Junior, II o Senior, III o Experto)
* No es posible salir del programa en cualquier momento. Si se solicita realizar una operación, debe terminarse para poder salir del programa.
* No existirán proyectos con el mismo nombre.

**Clases:**

Para realizar este proyecto crearemos un total de 7 clases, las cuales se comunicarán entre sí para lograr el correcto funcionamiento del programa. A continuación detallamos sus nombres y explicamos el funcionamiento y el rol en el programa de cada una de ellas:

* **ProgramaPrincipal:** Esta será la clase principal. A través de ella se podrá acceder a las operaciones solicitadas en el proyecto, mediante la comunicación con el resto de las clases. Aquí se declararan las principales variables del programa y se instanciarán los objetos de las clases restantes.
* **ConsultoraInformatica:** Estaclase tendrá como único atributo una lista de Proyectos, además de su constructor y un método que permita recorrer esa lista y acceder a los métodos específicos que se soliciten.
* **Proyecto:** Esta clase contendrá, junto con sus *getters*, *setters* y su respectivo constructor(al igual que el resto de las clases) los siguientes atributos: Nombre, fecha de inicio, fecha de fin, líder a cargo, monto de los gastos operativos y una lista que guardara la información de cada personal que participa del proyecto. Además, en esta clase se crearán métodos que permitirán ejecutar las operaciones correspondientes al proyecto.
* **Personal:** Esta clase contendrá los atributos DNI, nombre, apellido y cantidad de horas trabajadas por cada participante o personal del proyecto. Será en realidad una Superclase, de la que dependerán dos Subclases que serán descriptas a continuación:
  + - **Desarrollador:** Además de los atributos de su Superclase, contendrá los atributos: Lenguaje (indica en qué lenguaje se especifica) y Nivel (I, II, o III), junto con su métodos específicos.
    - **Tester:** No tendrá ningún atributo adicional, además de su constructor y los métodos especiales que se piden.

Además, la clase Personal contara con dos métodos abstractos cuya función será invocar a los métodos que contengan su mismo nombre en las subclases, para ejecutar sus instrucciones de manera diferente, según pertenezcan a la clase Desarrollador o Tester.

* **Lista:** Esta clase, como su nombre lo indica, será utilizada para crear una lista a partir de la información ingresada en el programa, tanto de proyectos como de participantes del mismo.

**Operaciones:**

A continuación explicaremos de qué manera implementaremos el código del programa para poder ejecutar las 4 operaciones que se solicitan:

* **Agregar y modificar datos de los proyectos:** Para lograr esta operación simplemente implementaremos los *getters* y *setters* propios de la clase Proyecto, para así poder agregar o modificar cada uno de sus datos. Para poder “localizar” el proyecto que se quiere modificar, recorreremos la lista de proyectos que ya ha sido implementada hasta encontrar el que se está buscando.
* **Crear y agregar participantes para un proyecto en particular:** Para esta operación, se recorrerá y buscará en la lista el proyecto al que se le quieren agregar participantes, y si es encontrado se invocará un método que se encuentra en la clase Proyecto que permita leer todos los datos del nuevo participante y luego lo agregue a la lista correspondiente.
* **Imprimir la información de los participantes de un proyecto:** En este caso también utilizaremos los métodos para recorrer la lista de proyectos y poder encontrar el proyecto buscado. Una vez encontrado, se invocará a un método de la clase Proyecto que permita recorrer la lista de participantes, y que a medida que lo hace, invoque al método abstracto de la clase Personal que imprimirá en pantalla los datos correspondientes a sus dos subclases.
* **Calcular el costo de un proyecto con la información almacenada:** Esta operación se implementará de manera similar a la anterior, con la diferencia de que al recorrer la lista de participantes, en lugar de invocar al método que permite imprimir los datos de éstos, llamará al método (también abstracto) que realice las operaciones correspondientes para devolver la suma del costo de cada uno los participantes del proyecto, y que al finalizar el recorrido se le agregue a este valor el monto por gastos operativos del proyecto, para finalmente poder informar el costo final del proyecto buscado.