World.h

```
class World{
    private: //atributos
        Simulator * simulator;
    public: //metodos
        World(int rows,int columns,int people,
            double perCent_infected);
        ~World();
        void passDay();
        friend ostream & operator << (ostream
                        &out,const World &w)
};
```

Simulator.h

```
class Simulator{
   //atributos
   vector<Country*> list_countries;
   int days_passed;
   int rows;
   int columns;
   public: //metodos
       Simulator(int r,int c);
       void populate(int peop,double pcinf);
       ~Simulator();
       void passDay();
       int getDaysPassed();
       void westNeighbourAdd(Country *  c, int index);
       void eastNeighbourAdd(Country * c, int index);
       void northNeighbourAdd(Country * c, int index);
       void southNeighbourAdd(Country * c, int index);
      friend ostream & operator << (ostream &out, const
                                    Simulator * sim);
};
```

SimulatorRules.h

```
class SimulationParameters{
    //atributos
    private:
        int daysMaxStay =  10;
        int daysMinStay = 5;
        int daysUntilSick = 6;
        int daysUntilDeadChance = 8;
        int daysUntilImmune = 10;
        int daysUntilHealthy = 2;
        //probabilidades de cosas
        double probToTransmitVirus = 0.4;
        double probToDie = 0.25;
    //metodos
    public:
        SimulationParameters();
        ~SimulationParameters();
        int getMaxStayDays();
        int getMinStayDays();
        int getUntilSickDays();
        int getUntilDeadChanceDays();
        int getUntilImmuneDays();
        int getUntilHealthyDays();
        bool infectionDiceThrow();
        bool travelDiceThrow();
        bool dieDiceThrow();
};
extern SimulationParameters g_simpars;
```