



Universidad  
Nacional  
de Quilmes

---

-->TEQ<--

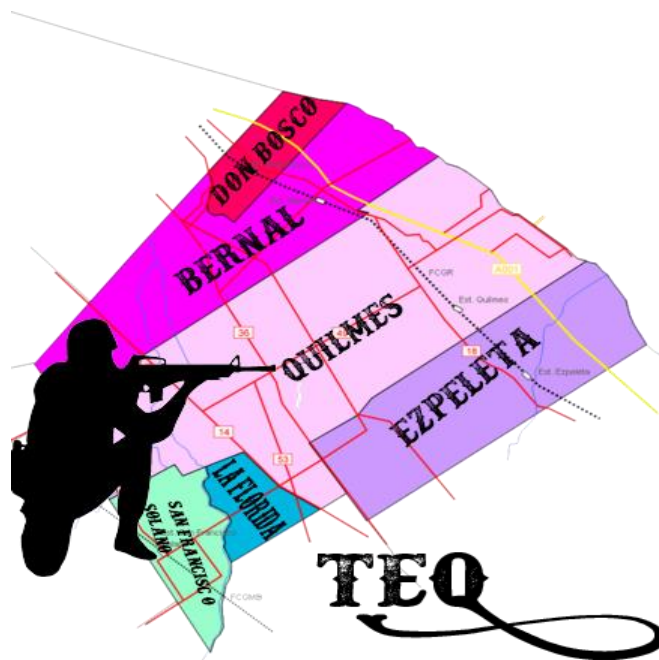
---

## PROGRAMACIÓN CON OBJETOS 1

GRUPO 8

**INTEGRANTES:** Lautaro Cornaglia y Flavio Tolaba.

**FECHA DE ENTREGA:** 10/12/2020



En el presente trabajo se han implementado todas las partes del enunciado, algunas con pequeñas modificaciones. Por ejemplo; en el enunciado se menciona que TEQ tiene 9 localidades y se ganaba cuando un jugador consiguiera 6 localidades; cuando realmente Quilmes solo tiene 6 localidades, esto llevo a cambiar el modo en que ganaba un jugador. Ahora un jugador gana cuando posee 4 localidades.

#### **Características adicionales:**

1. TEQ tiene 6 localidades y no 9.
2. Un jugador gana cuando tiene 4 localidades y no 6.
3. Si bien TEQ entiende el mensaje agregarJugador: XJugador , no es necesario ya que no se pueden agregar jugadores en medio de una partida. Por eso lo instanciamos desde un inicio en el setUp con los jugadores que vayan a jugar.
4. Pensamos en hacer un mensaje para TEQ que sea repartirTurnos , pero no era necesario, ya que al instanciar al TEQ en el setUp nosotros le damos el orden en el que serán los turnos.
5. El reparto de localidades y fichas se empieza desde la segunda posición de los jugadores, debido a que usamos un **CircularReadStream**

## **MODELO**

Los objetos más importantes de nuestro sistema son:

- **TEQ:** es nuestra plataforma donde se iniciara este juego. Sus principales responsabilidades son:
  1. Que pueda saber las localidades que tiene para repartir entre sus jugadores.
  2. Que sepa cuantos jugadores tiene para jugar.
  3. Se encarga de realizar el duelo entre dos jugadores y las localidades correspondientes
  4. Se encarga de tomar las validaciones para que un jugador pueda atacar. Por ejemplo; que la localidad atacante del jugador que ataca tenga 2 o más tropas para atacar.
  5. Se encarga de iniciar la fase de enrolamiento.
  6. Desde el initialize de la partida, es decir el TEQ, se realiza el reparto de localidades y se ponen las fichas correspondientes.
- **Localidad:** es un objeto esencial de nuestro sistema ya que la localidad es la que va a determinar si un jugador pueda atacar o no, la localidad sabe si es limítrofe o no a otras localidades, sabe la cantidad de ejércitos que tiene y de que color de jugador son.
- **Jugador:** es el objeto con el que vamos a realizar ataques, determinar si hay un ganador en la partida y diferentes validaciones que tiene que tener el jugador para poder atacar;el también sabe cuantas localidades tiene y cuales son sus localidades.
- **TiradorDeDados:** es el objeto que nos dará los números de dados para que los jugadores puedan realizar sus combates entre ellos, este objeto es necesario ya que seria imposible

testear una tirada de dados con números random, lo cual es posible con el mensaje `atRandom`:

- **ResolucionDeCombate**: es la clase abstracta de **GanaAtacante** y **GanaDefensor** los cuales se determinara en el método del jugador al realizar el ataque, tomara alguno de esos dos casos y dará el resultado correspondiente.

## PROGRESO

El TP fue construido de igual manera que otras actividades, empezando con el test más sencillo y haciendo que los test rompan para crear los mensajes que correspondan a cada objeto (TDD).

Partimos realizando algunos test que nos devolvieran información de la partida de TEQ, por ejemplo; la cantidad de jugadores que tenía la partida, la cantidad de localidades que tenía un jugador y de que color eran las fichas que tenía cada localidad; y luego empezamos a realizar los test que cambiaban el estado del TEQ como por ejemplo el sistema de combate. Luego de realizar el sistema de combate llegamos a la fase de enrolamiento, que fue una tarea que se realizó bastante rápido y por último se realizó el uso de Streams para imprimir diferentes estados de partida, por ejemplo: la partida inicial del TEQ instanciada en el `setUp`, la partida de TEQ luego de realizar un combate y también una partida de TEQ de 4 jugadores que fue realizada con colaboradores temporales.

### **DIFICULTADES:**

En la parte inicial del TP, que fue crear los objetos y probar los test lo hicimos bastante rápido; la primera traba llegó al momento de repartir las localidades, dependiendo de la cantidad de jugadores que tenga la partida; por ejemplo si hay 2 jugadores en el TEQ, el juego repartirá 3 localidades para cada uno; si son 3 jugadores el juego repartirá 2 localidades para cada uno y por último si son 4 jugadores, el juego repartirá de manera dispareja las localidades entre los jugadores.

La segunda traba fue realizar el sistema de combate entre jugadores, ya que tuvimos dificultades al momento de realizar los diferentes casos de combate, al final pudimos resolverlo y mejorar bastante el código gracias a las consultas realizadas a los docentes.

### **CONCLUSIÓN**

Disfrutamos mucho realizando este TP ya que pudimos aplicar todos los conceptos que aprendimos en la cursada (Streams, proveedor de Estados, manejo de clases y excepciones) también pudimos ganar experiencia trabajando en conjunto ya que realizábamos llamadas donde intercambiábamos ideas para llegar a una resolución del TP mucho más correcta.