

Desarrollo de software

1.5 Métodos Ágiles

3.1 Organización de equipo

Rol	Responsabilidad
Scrum Master	Vigila que el equipo siga los pasos del sprint y resuelve bloqueos
Product Owner	Prioriza las tareas y define el objetivo
Equipo de desarrollo	Programan y prueban el código

3.2 Reuniones en SCRUM

Rol	Alumnos	Descripción
Scrum Master	David López	Facilita el trabajo en equipo, ayuda a resolver problemas y asegura que se sigan las reglas de Scrum
Product Owner	David Wei Xu	Elige el programa que desarrollar
Desarrollador/a	David López David Wei Xu	Escribe el código, colabora con los demás y prueba el programa

- Fase 1 - Sprint Planning (5 min):
 - El Product Owner elige y presenta el objetivo del sprint al equipo: puede seleccionar uno de esos programas o inventar otro.
[Simulador de dado: Lanza un dado y muestra el resultado aleatorio.](#)
 - Luego, todo el equipo decide qué tareas se realizarán para realizar el programa y las anotan en el Product Backlog:

//	Tareas del Sprint (Product Backlog)
1	Crear el programa en Java
2	Pruebas
3	Resolver problemas de ser necesario
4	Enviar al cliente

- ¿Cómo ha decidido el equipo qué tareas son las más importantes?
[Mediante debate y siguiendo un orden cronológico que atienda a los posibles problemas](#)

- Fase 2 - Trabajo del Sprint (10 min):
 - Durante esta fase, los desarrolladores programan el pequeño proyecto en Python o Java (aquí todo el equipo puede ayudar!)

```
package Entornos_de_desarrollo.Ejercicios_resueltos;

import java.util.Scanner;

public class Ejercicio_Metodos_Agiles {
    public static void main(String[] args) {

        //Scanner y declaracion de variables
        Scanner input = new Scanner(System.in);
        int numUser, numDice, attempt = 3, eleccion;

        //Eleccion de modo de juego
        System.out.print("Choose a game mode:      \n1. Roll the dice 3 times and guess
the number each time, 1 attempt per roll  \n2. Roll the dice 1 time and guess the
number (3 attempts)      \nYour choice: ");
        eleccion = input.nextInt();

        switch (eleccion) {
            case 1:
                //Salida de datos
                while (attempt>0){

                    //Entrada de datos
                    System.out.print("Guess the number between 1 and 6 (You have " +
attempt + " attempts): ");
                    numUser = input.nextInt();

                    //Generacion de numero aleatorio
                    numDice = (int) (Math.random() * 1) + 1;

                    if (numUser == numDice) {
                        System.out.println("Congratulations! You guessed the number " +
numDice);
                        break;
                    } else {
                        System.out.println("Sorry, the correct number was " + numDice);
                        attempt--;
                    }
                }
                if (attempt == 0){
                    System.out.println("You've used all your attempts. Better luck next
time!");
                }
                break;
        }
    }
}
```

```

        case 2:
            //Entrada de datos
            System.out.print("Guess the number between 1 and 6 (You have " +
attempt + " attempts): ");
            numUser = input.nextInt();

            //Generacion de numero aleatorio
            numDice = (int)(Math.random() * 6) + 1;
            attempt--;

            //Salida de datos
            while (attempt>0){

                //Condicionales
                if (numUser == numDice) {
                    System.out.println("Congratulations! You guessed the number " +
numDice);

                    break;
                } else {
                    System.out.print("Sorry, that was incorrect. Try again. You
have " + attempt + " attempts left: ");
                    numUser = input.nextInt();
                    attempt--;
                }
                if (attempt == 0){
                    System.out.println("You've used all your attempts. Better luck next
time! The correct number was " + numDice);}

            }

            break;

        default:
            System.out.println("You must enter the number 1 or 2 to choose the game
mode. Not other options are available.");
    }

    //Cierre del Scanner
    input.close();
}
}

```

- Fase 3 - Daily Scrum (5 min) :
 - Simulad una breve reunión de equipo (máximo 15 minutos en la vida real, aquí sólo 5).
 - El equipo responde a tres preguntas:
 - 1. ¿Qué parte del programa he creado?
Hemos creado el programa simultáneamente
 - 2. ¿Qué parte del programa podría hacer o mejorar ahora?
Hemos creado otro modo de juego dentro para mayor variedad
 - 3. ¿Qué problema he tenido o tengo ahora?
Hemos tenido un problema con el número de attempts que no bajaba cuando fallabas, ya está solucionado
- Fase 4 - Sprint Review (5 min):
 - Cada equipo presenta su programa a otro grupo de la clase.
Se lo hemos presentado al grupo de Rodrigo
 - El otro grupo actuará como cliente y dará feedback.
Muy buena feedback, lo único hubo un error de asignación de modo de juego, que ya lo hemos corregido
- Fase 5 - Sprint Retrospective (5 min):
 - El equipo reflexiona sobre el proceso del sprint:
 - ¿Qué funcionó bien en el equipo?
Hubo buena comunicación, entendimiento, cooperación e implicación en el trabajo.
 - ¿Qué se puede mejorar?
Prestar atención a los pequeños detalles del código.
 - ¿Crees que Scrum ayuda a organizar mejor el trabajo en grupo? ¿Por qué?
En nuestro caso no ha servido de mucha ayuda porque al ser un proyecto tan pequeño y simple, pero para proyectos más complejos y grandes, con más participantes, este modelo ayuda mucho.