TESTING DE SOFTWARE

PRACTICA 1: AGENDA ANUAL

Laura Casas Torres

José Antonio Bernal

A continuación se explican brevemente los errores encontrados gracias a cada uno de los tests y como han sido resueltos en el código original.

CLASE DIABREAK (DIA)

- TestDiaBreak(): se ha hecho para comprobar que la clase Dia cumple con los requisitos de implementación.
    - TestOk1 comprueba que no se pueda introducir 0 como día válido para la cita.
    - TestOk2 comprueba que no se pueda introducir 366 como día válido para la cita.
    - TestOk3 comprueba que tanto los días 1 como 365 se puedan introducir como días válidos para la cita.

  El constructor de la clase contemplaba el día 0 como válido y el día 365 como no válido lo que no cumplía con los requisitos de implementación. El código ha sido modificado de la siguiente forma para poder resolver dichos errores.
  CODIGO ANTIGUO:
  `if (diaNumero < 0 || diaNumero >= 365)`
  CODIGO NUEVO:
  `if (diaNumero <= 0 || diaNumero > 365)`

- TestBuscaSlot():
  El código anterior no pasaba ninguno de los test ya que no modificaba el array de citas en ningún momento. Por esto se ha añadido este bucle for que modifica si hay espacio suficiente para la cita los slots.

  ```
  for(int i = 0; i < duracion; i++) {
          citas[i] = new Cita("", duracion);
  }
  ```

  Se ha modificado la siguiente comparación *numSlots < MAX_CITAS_POR_DIA* añadiendole un = (*numSlots <= MAX_CITAS_POR_DIA*) para que se pudiesen reservar sesiones de ocho horas que es el máximo permitido.

  Además se ha añadido esta comparación `if(duracion > 0)` para que no fuese posible reservar sesiones de 0 horas.

- TestGetCita(): No ha sido necesario cambiar el código original al no haber encontrado errores.

- TestMuestraCita(): El test nos mostró que el mensaje devuelto cuando la hora no era valida era incorrecto. Este error ha sido modificado en el código original añadiendo "Hora no valida" en lugar de "Hora valida";

- TestValidaHora(): No ha sido necesario cambiar el código original al no haber encontrado errores.

- TestHuecoLibre(): El test nos mostró que el código no contemplaba la posibilidad de que se buscase un hueco libre a una hora fuera del rango. Esto ha sido arreglado en el código añadiendo la siguiente condición

```
If(horaIni >= 0 && horaIni <= 7)
```

- TestAsignarCita(): No ha sido necesario corregir el código original al no haber encontrado errores.

CLASE SEMANAREAK (SEMANA)

- TestSemanaBreak(): El test al constructor de la case nos ha mostrado que el código no tenía en cuenta que la última semana del año sería de un solo día. Hemos añadido la siguiente condición para que al introducir la semana 52 no falle.

```
if(diaDelAnio== 365) {
   dias[0] = new DiaBreak(diaDelAnio);
}
```

- TestDiaSemana(): El test nos mostró que el quinto día de la semana no devolvía viernes, se ha modificado el código añadiendo un break; en el case 4 perteneciente al viernes.

- TestGetNumeroSemana(): el test no mostró ningún error, no ha sido necesario corregir el código.

- TestGetDia(): El test nos mostró que los días no empezaban según el código implementado en 1 siendo este el Lunes, 2 siendo el Martes etc, si no que se el Lunes se consideraba el día 0 , siendo así el Sabado el día 5 y el Domingo el día 6. Se ha modificado el código cambiando la condición del ify restandole uno al dia de la semana que entra como parámetro.

CODIGO ANTIGUO:

```
if(diaSemana >= 1 && diaSemana <= DIAS_RESERVABLES)
```

CODIGO NUEVO:

```
diaSemana--;
if(diaSemana >= 0 && diaSemana < DIAS_RESERVABLES)
```

- TestPrimerHueco(): El test nos mostró que el código comenzaba las horas a partir de la hora cero y no a partir de la hora nueve como debería, para arreglarlose ha añadido el siguiente código dentro del condicional

CODIGO VIEJO

```
if (hueco!=-1)
  {
        disponible= diaSemana(dia) + " " + hueco + ":00";
        return disponible;
   }
```

CODIGO NUEVO

```
if (hueco!=-1)
   {
        hueco = hueco + 9;
        disponible= diaSemana(dia) + " " + hueco + ":00";
        return disponible;
    }
```

Además el test nos mostró que el primer día que el código contemplaba era el Martes puesto que el bucle comenzaba en el día 1, el código ha sido modificado para que el bucle comience en el dia 0 -> Lunes

```
for(int dia = 0; dia < DIAS_RESERVABLES; dia++)
```

- TestMostrarCita(): no se ha encontrado ningún error en el código.