



Proyecto 2:

Extensión del modelo e implementación de una versión sobre simplificada de un juego.

Caso de estudio: Candy Crush Saga.

Considerando los requisitos y la implementación previamente llevada a cabo para el **Proyecto Nº 1**, el presente proyecto consiste en el rediseño, extensión e implementación de una nueva versión sobre simplificada e inspirada en *Candy Crush*. En esta extensión del proyecto original, **se deben respetar e incorporar todas y cada una de las siguientes características:**

Patrones de diseño:

- utilizar **al menos tres patrones de diseño** (todos diferentes, sin importar el tipo), que deberán ser seleccionados en concordancia con el diagrama de clases obtenido. El uso de los tres patrones de diseño debe estar debidamente justificado.

Refactoring:

- la versión extendida **debe permitir que el jugador seleccione**, al iniciar la aplicación, entre **al menos dos dominios de aplicación** para visualizar el juego. En este sentido, podrá existir un dominio de aplicación como el original (*Caramelos*), y al menos uno adicional, a libre elección. La lógica aplicada a la aplicación es la misma, sin importar el dominio de aplicación.
- la versión extendida **debe incorporar el conteo de la puntuación** del jugador, que quedará asociada con la detonación de los diferentes caramelos tal como se describe en la *Tabla 1*.
- la versión extendida **debe incorporar dos tipos especiales de caramelos**, tal como se los describe en la *Tabla 2*.
- la versión extendida **debe permitir que**, junto con la descripción de cada nivel (indicada mediante el archivo de texto que se *parsea* al iniciar un determinado nivel) **se indique cuáles son las reglas de match que se aplicarán** sobre dicho nivel. Las reglas de match a seleccionar para cada nivel serán las que considera el juego originalmente. De esta forma, para un nivel podrá especificarse que el chequeo de *match* se realiza observando la regla *tres o más en línea* y, además, la estrategia *formación de Ts*, mientras que, para otros niveles, solo podría estar indicado el chequeo de *tres o más en línea*. Independientemente de la o las reglas para un nivel, asuma que, la configuración de cada nivel mediante el archivo de texto, indicará **al menos una regla** de aplicación obligatoria.

Persistencia:

- la versión extendida **debe permitir almacenar de forma persistente el ranking de los cinco mejores puntajes** obtenidos a lo largo del tiempo. Para esto, cada vez que un jugador finalice la partida, se deberá considerar el puntaje que haya obtenido y almacenarlo, en el caso que corresponda. El puntaje siempre quedará asociado al nombre del jugador que será oportunamente consultado. La interfaz gráfica **debe posibilitar consultar el ranking en cualquier momento**.



Universidad Nacional del Sur
Departamento de Ciencias e Ingeniería de la Computación
Tecnología de Programación (TdP)
Segundo Cuatrimestre 2023



Tabla 1: Puntajes asociados a la detonación de los diferentes tipos de caramelos del juego.

Caramelo	Puntaje
Rojo	5
Naranja	15
Amarillo	20
Verde	10
Azul	20
Púrpura	25
Rayado horizontal (cualquier color)	45
Rayado vertical (cualquier color)	35
Envuelto (cualquier color)	50
Glaseado	25
Gelatina (con cualquier caramelo)	10 + puntaje caramelo interno
Caramelo TdP-1	100
Caramelo TdP-2	150

Tabla 2: Nuevos caramelos del juego.

Caramelo	Color	
TdP-1	Todos	Apariencia: personalizarlo a libre elección, pero de forma clara y distinguible. Efecto: eliminar la fila y columna completa de caramelos en la que se encuentra. Detonación: además de detonar ante cualquier match de 3 o más según las reglas del juego, el caramelo detona cuando es intercambiado con cualquier otro caramelo de su mismo color, sin la necesidad de que el intercambio de lugar a un match 3. Formación: este tipo de caramelo puede ingresar como cualquier otro caramelo al tablero, esto es, porque forma parte del nivel, o porque reemplaza a un caramelo anterior ante la detonación del mismo.
TdP-2	-	Apariencia: personalizarlo a libre elección, pero de forma clara y distinguible. Efecto: finaliza la partida del jugador, independientemente de la cantidad de vidas. Este efecto se origina después de una X cantidad de segundos que transcurren de forma descendente y se indican mediante la imagen asociada al caramelo especial. Detonación: la detonación se da en el mismo sentido que los caramelos Glaseados, esto es, en consecuencia de la detonación caramelo adyacente que no es un Glaseado. Formación: este tipo de caramelos forman parte únicamente de algunos niveles, que pueden considerarse desafíos, donde el tablero se conforma de manera tal de, desafiar al jugador a desactivar los caramelos TdP-2 presentes, antes de perder. Este tipo de caramelo, por ejemplo, puede pensarse para niveles especiales y en consonancia con la cantidad de movimientos disponibles.



Universidad Nacional del Sur
Departamento de Ciencias e Ingeniería de la Computación
Tecnología de Programación (TdP)
Segundo Cuatrimestre 2023



A partir de la descripción de arriba, resuelva:

Ejercicio 1: Mediante alguna herramienta de modelado, redefina el diagrama de clases en UML en donde se puedan observar las clases que modelan el sistema a implementar, así como sus relaciones, atributos y servicios. Para esto considere de forma separada un diagrama de clases reducido, y otro diagrama de clases extendido. Exporte los diagramas en formato imagen o pdf y suba el/los archivos dentro de una carpeta denominada *Diagramas-UML* al repositorio *p2-comision-xx* (donde *xx* corresponderá con su número de comisión) que creará la cátedra y lo asignará como colaborador. Se esperan obligatoriamente **al menos seis archivos**, tres donde se encuentran los *diagramas reducidos de clases*, y tres para los *diagramas completos de clases*. Por cada acto de defensa asociado a este ejercicio (ver más adelante), deberán incorporar un *diagrama reducido* y *extendido* de clases y, finalmente, un *diagrama reducido y extendido* de clases final, con la entrega final del proyecto: de aquí, los seis archivos en total.

Ejercicio 2: A partir del modelo definido en el *Ejercicio 1*, e incorporando y considerando las observaciones indicadas por el ayudante designado para la corrección de su comisión, implemente completamente una versión del juego en Java haciendo uso de Java Swing. La versión deberá cumplir con cada uno de los siguientes ítems:

- Implementar completamente la lógica del juego.
- Desarrollar una interfaz adecuada para el desarrollo del juego.
- Los objetos para completar cada celda de la GUI deberán mostrarse basándose en algún tipo de imagen.

Ejercicio 3: Suba al repositorio *p2-comision-xx* (donde *xx* corresponderá con su número de comisión) que creará la cátedra y lo asignará como colaborador, el proyecto Java mediante el cual implementó todo lo solicitado en el *Ejercicio 2*. Los archivos subidos al repositorio deben permitir el correcto funcionamiento y visualización del proyecto completo. Asegúrese de no olvidar ningún archivo. El proyecto debe compilar, caso contrario se considerará *Desaprobado*. Es **condición de aprobación** que todos los integrantes de la comisión hayan trabajado sobre la concreción del proyecto. Para esto, la cátedra se reserva el derecho de analizar los *insights* de cada repositorio, a fin de establecer el nivel de participación de cada usuario.

Condiciones de entrega:

- El proyecto deberá resolverse en grupos, según las [comisiones de cursado](#) indicadas en el aula virtual de la materia en Moodle. La copia o plagio del proyecto es una falta grave. Quien incurra en estos actos de deshonestidad académica, desaprobará automáticamente el proyecto.
- El proyecto será evaluado en **tres etapas**. Dos de estas etapas **serán presenciales** y los alumnos deberán presentarse a clase para realizar una **defensa** del mismo.
- Entrega y evaluación del proyecto:**
 - Etapla 1: considera el análisis y la concreción parcial de los ejercicios 1 y 2.
 - Etapla 2: considera el análisis y la concreción parcial de los ejercicios 1 y 2.
 - Etapla 3: considera la concreción del ejercicio 3.
- Las **fechas límite de entrega y evaluación** de cada etapa, seguirán el siguiente formato:
 - Etapla 1: la primera fecha de puesta en común sobre el **diagrama de clases reducido y extendido** se realizará la **semana del 31 de octubre y 02 de noviembre**, entre las **10:00 y 12:00 hs. (#1)**



Universidad Nacional del Sur
Departamento de Ciencias e Ingeniería de la Computación
Tecnología de Programación (TdP)
Segundo Cuatrimestre 2023



- Etapa 2: la segunda fecha de puesta en común sobre el **diagrama de clases reducido y extendido** y el avance sobre implementación se realizará la **semana del 07 y 09 de noviembre**, entre las **10:00 y 12:00 hs. (#1)**
- Etapa 3: la fecha límite para subir la última versión considerable para corrección al repositorio operará el **viernes 24/11**, a las **23:59 hs.** Todo *commit* realizado con posterioridad a esta fecha y hora no será considerado para la evaluación del proyecto.

(#1) Cada comisión tendrá un turno asignado en el cual todos los integrantes deberán asistir a clases con el ayudante asignado para la corrección del proyecto, y donde se debatirá sobre los diagramas confeccionados, observando los detalles a corregir o mejorar. La asistencia a este espacio es obligatoria, y la evaluación realizada impactará en la nota individual de cada uno de los alumnos que integre la comisión. Los turnos se publicarán en el aula virtual de la materia en Moodle

- Los proyectos serán calificados según la siguiente escala: A, B y C corresponden a proyectos *Aprobados*; D corresponde a proyectos *Desaprobados*.
- Todo proyecto que no sea completado, entregado y defendido mediante las condiciones de entrega enunciadas, se considerará *Desaprobado*.