**PROGRAMACIÓN ORIENTADA A OBJETOS**

**Persistencia**

**2025-01**

**Laboratorio 6/6 [ :) ]**

# OBJETIVOS

1. Completar el código de un proyecto considerando requisitos funcionales.
2. Diseñar y construir los métodos básicos de manejo de archivos: abrir, guardar, importar y exportar.
3. Controlar las excepciones generadas al trabajar con archivos.
4. Experimentar las prácticas XP : Only one pair  [integrates code at a time](http://www.extremeprogramming.org/rules/sequential.html) .

Use [collective ownership](http://www.extremeprogramming.org/rules/collective.html).



# ENTREGA

 Incluyan en un archivo .zip los archivos correspondientes al laboratorio. El nombre debe ser los dos apellidos de los miembros del equipo ordenados alfabéticamente.

 Deben publicar el avance al final de la sesión y la versión definitiva en la fecha indicada en los espacios preparados para tal fin.

**DESARROLLO**

# Preparando

En este laboratorio vamos a extender el proyecto schelling adicionando un menú barra con las opciones básicas de entrada-salida y las opciones estándar nuevo y salir.

1. En su directorio descarguen la versión del proyecto realizado por ustedes para el laboratorio 03 y preparen el ambiente para trabajar desde **CONSOLA**
2. Ejecuten el programa, revisen la funcionalidad.

# Creando la maqueta

## [En lab06.doc, \*.asta y \*.java] [NO OLVIDEN BDD y MDD]

En este punto vamos a construir la maqueta correspondiente a esta extensión siguiendo el patrón MVC.

1. **MODELO:** Preparen en la clase fachada del dominio los métodos correspondientes a las cuatro opciones básicas de entrada-salida (open, save, import y export). Los métodos deben simplemente propagar una CityException con el mensaje de “Opción nombreOpción en construcción. Archivo nombreArchivo”. Los métodos deben tener un parámetro File.

**Texto

El contenido generado por IA puede ser incorrecto.**

1. **VISTA :** Construyan un menú barra que ofrezca, además de las opciones básicas de entrada-salida, las opciones estándar de nuevo y salir (Nuevo, Abrir, Guardar como, Importar, Exportar como, Salir). No olviden incluir los separadores. Para esto creen el método prepareElementsMenu. Capturen la pantalla del menú.

Gráfico, Gráfico de barras

El contenido generado por IA puede ser incorrecto.

1. **CONTROLADOR:** Construyan los oyentes correspondientes a las seis opciones. Para esto creen el método prepareActionsMenu y los métodos base del controlador (optionOpen, optionSave, optionImport, optionExport, optionNew, optionExit), En las opciones que lo requieran usen un FileChooser y atiendan la excepción. Estos métodos llaman el método correspondiente de la capa de dominio que por ahora sólo lanza una excepción.Ejecuten las diferentes acciones del menú y para cada una de ellas capture una pantalla significativa.

Gráfico

El contenido generado por IA puede ser incorrecto.Tabla

El contenido generado por IA puede ser incorrecto.Tabla

El contenido generado por IA puede ser incorrecto.

# Implementando salir y nuevo

## [En lab06.doc, \*.asta y \*.java] [NO OLVIDEN BDD y MDD]

Las opciones salir y nuevo van a ofrecer los dos servicios estándar de las aplicaciones. El primero no requiere ir a capa de dominio y el segundo sí.

1. Construyan el método optionExit que hace que se termine la aplicación. No es necesario incluir confirmación.

* En el editor de elección del lector.

A black screen with white text

AI-generated content may be incorrect.

1. Construyan el método optionNew que crea una nueva ciudad. Capturen una pantalla significativa.

* En el editor de elección del lector.

A computer screen shot of a black screen

AI-generated content may be incorrect.



# Implementando salvar y abrir

## [En lab06.doc, \*.asta y \*.java] [NO OLVIDEN BDD y MDD]

Las opciones salvar y abrir van a ofrecer servicios de persistencia de la ciudad como objeto. Los nombres de los archivos deben tener como extensión .dat.

1. Copien las versiones actuales de open y save y renómbrenlos como open00 y save00
2. Construyan el método save que ofrece el servicio de guardar en un archivo el estado actual de la ciudad. Por ahora para las excepciones sólo consideren un mensaje de error general. No olviden diseño y pruebas de unidad.

A computer code on a black background

AI-generated content may be incorrect.

A screen shot of a computer code

AI-generated content may be incorrect.

1. Validen este método guardando el estado obtenido después de dos clics como oneCity.dat. ¿El archivo se creó en el disco? ¿Cuánto espacio ocupa?

A screenshot of a computer

AI-generated content may be incorrect.

* Ocupa un total de 2kb.

1. Construyan el método open que ofrece el servicio de leer una ciudad de un archivo. Por ahora para las excepciones sólo consideren un mensaje de error general. No olviden diseño y pruebas de unidad.

A computer screen with colorful text

AI-generated content may be incorrect.

1. Realicen una prueba de aceptación para este método iniciando la aplicación, creando un nuevo estado y abriendo el archivo oneCity.dat. Capturen imágenes significativas de estos resultados.

A screen shot of a computer

AI-generated content may be incorrect.

# Implementando importar y exportar

## [En lab06.doc, \*.asta y \*.java] [NO OLVIDEN BDD y MDD]

Estas operaciones nos van a permitir importar información de la ciudad desde un archivo de texto y exportarla. Los nombres de los archivos de texto deben tener como extensión .txt

Los archivos texto tienen una línea de texto por cada elemento

En cada línea asociada un elemento se especifica el tipo y la posición.

Person 10 10

Walker 20 20

1. Copien las versiones actuales de import y export y renómbrenlos como import00 y export00
2. Construyan el método export que ofrece el servicio de exportar a un archivo texto, con el formato definido, el estado actual. Por ahora para las excepciones sólo consideren un mensaje de error general. No olviden diseño y pruebas de unidad.

A computer screen shot of text

AI-generated content may be incorrect.

1. Realicen una prueba de aceptación de este método: iniciando la aplicación y exportando como oneCity.txt. Editen el archivo y analicen los resultados. ¿Qué pasó?

El método export crea el archivo legible y editable, al modificar manualmente se guarda y la aplicación lo sobrescribe sin advertencia.

A screenshot of a computer

AI-generated content may be incorrect.

1. Construyan el método import que ofrece el servicio de importar de un archivo texto con el formato definido. Por ahora sólo considere un mensaje de error general. No olviden diseño y pruebas de unidad.

(Consulten en la clase String los métodos trim y split)

A screen shot of a computer code

AI-generated content may be incorrect.

1. Realicen una prueba de aceptación de este par de métodos: iniciando la aplicación exportando a oneCity.txt. saliendo, entrando, creando una nueva e importando el archivo otherCity.txt. ¿Qué resultado obtuvieron? Capturen la pantalla final.

Los métodos cumplen su propósito básico, pero requieren mayor robustez para manejar entradas manuales inválidas, casos extremos (valores negativos, formatos incorrectos).

A grid with yellow dots

AI-generated content may be incorrect.

1. Realicen otra prueba de aceptación de este método escribiendo un archivo de texto correcto en oneCity.txt. e importe este archivo. ¿Qué resultado obtuvieron? Capturen la pantalla.
2. A grid with yellow dots

   AI-generated content may be incorrect.

# Analizando comportamiento

## [En lab06.doc, \*.asta y \*.java] [NO OLVIDEN BDD y MDD]

1. Ejecuten la aplicación, den tres clics, salven a un archivo cualquiera y ábranlo. Describan el comportamiento

Al ejecutar la aplicación, dar tres clics (tic-tacs) y guardar con save00(), se crea un archivo binario .dat que contiene el estado exacto de la ciudad en ese momento, incluyendo agentes, sus posiciones, estados internos y el contador de tic-tacs. Al abrirlo con open00(), la ciudad se reconstruye *idénticamente* al momento del guardado, preservando incluso el número de tic-tacs ejecutados (ej: si se guardó en el tic-tac 3, al abrir se mantiene en 3). Los objetos se restauran mediante serialización Java, lo que garantiza fidelidad completa con el estado original, incluyendo referencias internas entre agentes y la ciudad.

1. Ejecuten la aplicación, tres clics, exporten a un archivo cualquiera e importen. Describan el comportamiento

Al exportar con export00() después de tres tic-tacs, se genera un archivo de texto (.txt) legible, pero con información *selectiva*. La importación recrea una ciudad *similar* pero no idéntica:

* + El contador de tic-tacs se reinicia a 0 (no se preserva el historial).
  + Los agentes se recrean en sus posiciones y estados finales, pero como instancias nuevas (pierden referencias a estados anteriores).
  + Los ítems no-agentes (como semáforos) se reinstancian, lo que puede afectar su estado interno (ej: un semáforo podría reiniciar su ciclo).

1. ¿Qué diferencias ven el comportamiento 1. y 2.? Expliquen los resultados.

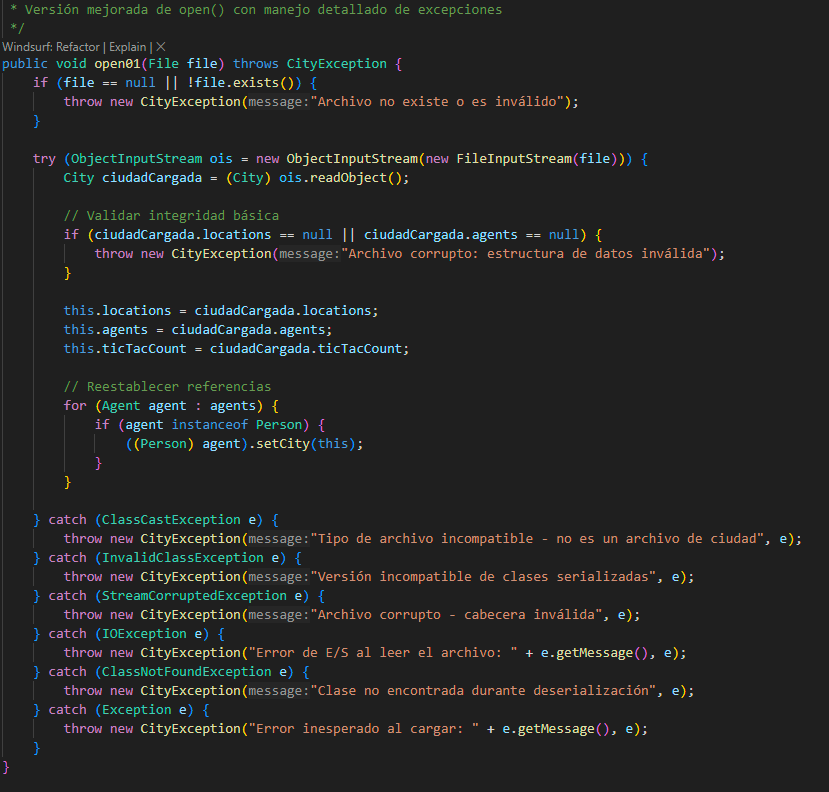
La principal diferencia radica en la *pérdida de estado interno* y *contexto temporal* en la versión texto (export/import). Mientras el .dat conserva toda la estructura de objetos serializados (incluyendo referencias y estados ocultos), el .txt solo almacena datos básicos (coordenadas, color, estado visible). Esto ocurre porque se realiza de dos maneras diferentes

* + Serialización binaria (save/open): Es un volcado completo de memoria. Ideal para recuperar estados exactos, pero menos portable y editable.
  + Texto plano (export/import): Es una reconstrucción manual. Más flexible para edición, pero pierde información no explícita (como el contador de tic-tacs o estados complejos de ítems).

# Perfeccionando salvar y abrir

## [En lab06.doc, \*.asta y \*.java] [NO OLVIDEN BDD y MDD]

1. Copien las versiones actuales de open y save y renómbrenlos como open01 y save01
2. Perfeccionen el manejo de excepciones de los métodos open y save detallando los errores. No olviden pruebas de unidad.
3. Realicen una prueba de aceptación para validar uno de los nuevos mensajes diseñados, ejecútenla y capturen la pantalla final. **Perfeccionando importar y exportar.**



A computer screen shot of text

AI-generated content may be incorrect.

## [En lab06.doc, \*.asta , cityErr.txt \*.java] [NO OLVIDEN BDD y MDD]

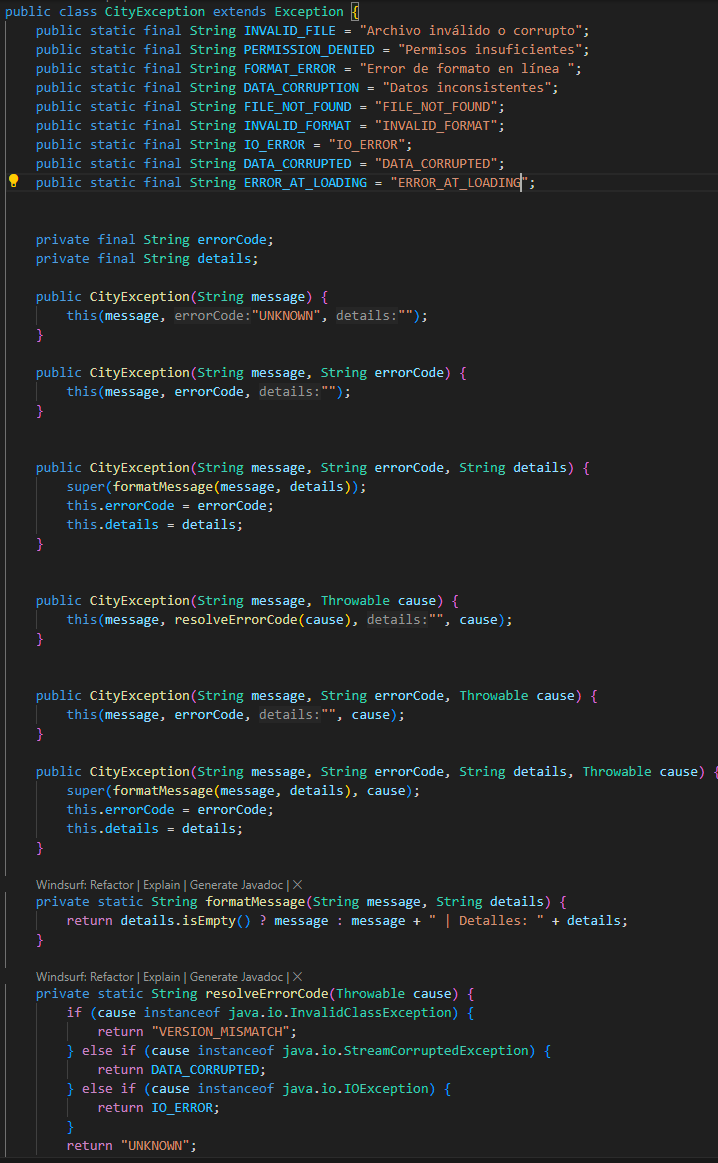
1. Copien las versiones actuales de import y export y renómbrenlos como import01 y export01
2. Perfeccionen el manejo de excepciones de los métodos import y export detallando los errores. No olviden pruebas de unidad.
3. Realicen una prueba de aceptación para validar uno de los nuevos mensajes diseñados, ejecútenla y capturen la pantalla final. **Perfeccionando importar. Hacia un minicompilador.**

A screen shot of a computer program

AI-generated content may be incorrect.

A screen shot of a computer program

AI-generated content may be incorrect.

Fue necesario Complementar la clase CityException de la siguiente manera

**RETROSPECTIVA**

1. **¿Cuál fue el tiempo total invertido en el laboratorio por cada uno de ustedes ?(Horas/Hombre)**

El tiempo total invertido en el laboratorio por cada uno de los integrantes fue de aproximadamente 10 horas/hombre.

1. **¿Cuál es el estado actual del laboratorio? ¿Por qué?**

El estado actual del laboratorio es funcional con mejoras pendientes. Se logró completar el desarrollo principal del sistema, y aunque cumple con los objetivos mínimos planteados, aún se pueden optimizar ciertos aspectos del código y refinar detalles en la interfaz o rendimiento. Esta situación se debe a que el enfoque principal fue lograr que el sistema funcionara correctamente antes de dedicar tiempo a mejoras adicionales.

1. **Considerando las prácticas XP del laboratorio. ¿cuál fue la más útil? ¿por qué?**

De las prácticas XP implementadas en el laboratorio, la más útil fue la programación en parejas (pair programming). Esta técnica permitió una revisión constante del código, una mejor distribución del conocimiento entre los miembros y una disminución significativa en la cantidad de errores. Además, facilitó la toma de decisiones rápidas.

1. **¿Cuál consideran fue el mayor logro? ¿Por qué?**

Consideramos que el mayor logro del laboratorio fue lograr una comunicación efectiva entre los distintos módulos del sistema, ya que este aspecto resultó clave para el funcionamiento global del proyecto. Lograr que los datos fluyeran correctamente entre componentes y que se procesaran según lo esperado representó una meta técnica importante alcanzada gracias al trabajo en equipo y la perseverancia.

1. **¿Cuál consideran que fue el mayor problema técnico? ¿Qué hicieron para resolverlo?**

El mayor problema técnico que enfrentamos fue la integración de los componentes de software con la entrada de datos en tiempo real, especialmente por incompatibilidades entre formatos y sincronización. Para resolverlo, realizamos una depuración exhaustiva del código, analizamos el flujo de datos paso a paso y modificamos la lógica de algunos módulos para adaptarlos mejor al entorno de ejecución. Esto nos permitió finalmente estabilizar el sistema.

1. **¿Qué hicieron bien como equipo? ¿Qué se comprometen a hacer para mejorar los resultados?**

Como equipo, hicimos bien en mantener una comunicación clara y constante durante todo el proceso, asignando responsabilidades de forma equitativa y respetando los tiempos acordados. Para mejorar nuestros resultados en futuros proyectos, nos comprometemos a comenzar la implementación con más anticipación, realizar pruebas más frecuentes y documentar mejor los cambios realizados para evitar redundancias o confusiones.

1. **¿Qué referencias usaron? ¿Cuál fue la más útil? Incluyan citas con estándares adecuados**.

Para el desarrollo del laboratorio, nos apoyamos de asistentes virtuales, que nos ayudaron a resolver dudas técnicas, aclarar conceptos y generar ideas para el desarrollo del código. No utilizamos libros físicos ni artículos académicos formales, sino que recurrimos a consultas en tiempo real con estas herramientas, las cuales resultaron muy útiles por su rapidez, claridad y capacidad de adaptarse a nuestras necesidades específicas durante el proceso de desarrollo.