**FIUBA - 75.07**

**Algoritmos y programación III**

*Trabajo práctico 2: AlgoCity*

2do cuatrimestre, 2014

(trabajo grupal)

Alumnos:

|  |  |  |
| --- | --- | --- |
| Nombre | Padrón | Mail |
| Andujar, Martin | 95099 | andujar06@gmail.com |
| Blanco, Joaquin | 94653 | joaquinbf@live.com |
| Bolaño, Nahuel | 94176 | naku07@gmail.com |
| Pringles, Elias | 95380 | elo94@outlook.com |

***Fecha de entrega final***: 19/12/2014

***Tutor***: Falcone, Gabriel

***Nota Final***:

# 

# 

**Introducción**

Aplicar los conceptos enseñados en la materia a la resolución de un problema, trabajando en forma grupal y utilizando un lenguaje de tipado estático (Java)

## Consigna general

Desarrollar la aplicación completa, incluyendo el modelo de clases e interface gráfica. La aplicación deberá ser acompañada por pruebas unitarias e integrales y documentación de diseño. En la siguiente sección se describe la aplicación a desarrollar.

# **Descripción de la aplicación a desarrollar**

AlgoCity es un juego ***similar*** al Famoso SimCity[[1]](#footnote-2). El mismo se basa en la construcción y administración de una ciudad.

Los edificios se dividen en:

* **Residenciales**: Aquí los ciudadanos construirán sus casas. Asegurarse de que disponen de electricidad, cloacas y acceso al tránsito.
  + Abarca una hectárea:
    - costo: $5
    - Capacidad máxima de alojamiento por hectárea: 100 personas.
    - Consumo eléctrico: 1MW
* **Comerciales**: Aquí los ciudadanos construirán tiendas comerciales. Asegurarse de que disponen de electricidad, cloacas y acceso al tránsito.
  + Abarca una hectárea
    - costo: $5
    - Consumo eléctrico: 2MW
* **Industriales**: Aquí los ciudadanos construirán sus empresas. Asegurarse de que disponen de electricidad y acceso al tránsito.
  + Abarca una hectárea:
    - costo: $10
    - Generan empleo hasta para 25 personas por hectárea
    - Consumo eléctrico: 5MW

Para conectar las hectáreas disponemos de:

* **Ruta pavimentada**: $10 por hectárea. Conectan las distintas zonas,
* **Líneas de Tensión**: $5 por hectárea. Son necesarias para proporcionar electricidad a cualquier zona. Sólo se unen entre sí y con destino a una Zona (Residencial, Comercial, Industrial) desde una central eléctrica. A ningún otro tipo de edificio.
* **Tubería de agua**: La misma transporta agua hasta una hectárea. Sólo se conecta desde un pozo de agua. $5 por hectárea.

Centrales eléctricas (necesitan estar conectados a la red de agua para funcionar) :

* **Central eólica**: ( $1000. costo de construcción) - Proporcionan electricidad a la zona circundante en un radio de 4 hectáreas sin necesidad de líneas de tensión en ese rango.
  + Capacidad de abastecimiento: 100 MW
* **Central Mineral**: ( $3000 costo de construcción) - Proporcionan electricidad a la zona circundante en un radio de 10 hectáreas sin necesidad de líneas de tensión en ese rango.
  + Capacidad de abastecimiento: 400 MW
* **Central Nuclear**: ( $10000 costo de construcción) - Proporcionan electricidad a la zona circundante en un radio de 25 hectáreas sin necesidad de líneas de tensión en ese rango.
  + Capacidad de abastecimiento: 1000 MW

Superficies:

* **Terreno llano**: El terreno se encuentra apto para comenzar a construir sin ningún costo.
* **Agua**: Sólo los edificios habilitados se pueden construir sobre el agua. (Pozo de agua y tuberías de agua)

**Pozo de agua**: ( $250 costo de construcción ) Este edificio debe construirse sobre una superficie de agua y a través de las tuberías proporciona agua a las hectáreas.

**Estación de bomberos**: ($1500 costo de construcción) Son los encargados de reparar los edificios que hayan sido afectados por alguna catástrofe. Poseen una velocidad de reparación:

* edificios residenciales: 10% de reconstrucción por cada turno.
* edificios comerciales 7% de reconstrucción por cada turno.
* edificios industriales: 3% de reconstrucción por cada turno.
* centrales eléctricas:
  + Eólicas: 15% por turno.
  + Minerales: 10% por turno
  + Nuclear: 3% por turno.

**Catástrofes**

Durante la partida pueden ocurrir las siguientes catástrofes:

* **Terremoto**: Se origina en un punto aleatorio del mapa y causa un daño proporcional a la distancia en que se encuentre el edificio impactado.(Decrece un 1,5% a medida que se aleja una hectárea de distancia).

* **Godzilla**: Se origina en un punto aleatorio del borde del mapa y desaparece en el lugar opuesto según su manera de caminar. Godzilla, según se despierte decide caminar en línea recta, haciendo zig-zag o en diagonal. Causa un daño de la siguiente manera:

|  |  |
| --- | --- |
| **Daños provocados por Godzilla** | **Formas en que se puede mover Godzilla** |
| * Residenciales: 100% * Comerciales: 75% * Industriales: 40% * Líneas de Tensión: 100%. * Ruta: 80% * Centrales eléctricas: 35% | movimiento godzilla.png |

**Turnos**

El tiempo (cada turno) en el juego pasa en períodos discretos. Luego de 30 turnos los ciudadanos pagan $10 x ciudadano.

**Jugadores**

La aplicación debe soportar múltiples jugadores (no simultáneos). Al iniciar el juego el sistema debe consultar al usuarios si ya posee uno o si desea crear uno nuevo.

*Validaciones*:

* Nombre de usuario debe contener por lo menos 4 caracteres.
* No puede haber nombres de usuarios repetidos.

**Persistencia**

La aplicación debe tener la funcionalidad de poder guardar (y recuperar) el estado actual del juego por cada jugador existente. Aún no se ha decidido si la misma será en formato XML o formato JSON.

**Nota**: Sea cauteloso. Realice el diseño que le parezca más adecuado a los efectos de minimizar lo más posible la cantidad de trabajo al momento en que se reciba la información sobre el tipo de archivo a utilizar para persistir.

**Interfaz gráfica**

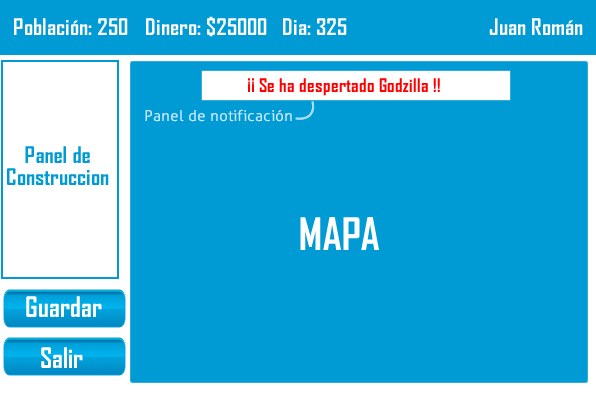
La misma debe ser clara y sencilla facilitando la jugabilidad al usuario. En una pantalla inicial al abrirse el juego debe indicar claramente el nombre del juego y una imagen de fondo que sea representativa del mismo. Los puntajes se calculan en base a la cantidad de población de la ciudad de cada jugador.







**Nota**: La pantalla a continuación se puede utilizar tanto para una nueva partida como para retomar las guardadas.



**Atención**: Cada vez que se desee construir tuberías de agua, el mapa debe cambiar a su forma “Visión subterránea” en donde sólo se ve la red de tubería. En la visión de la superficie del mapa NO deberían verse las tubería, pero sí los pozos de agua.

**Atención II**: Recuerde que la vista debe representar el estado actual de un objeto. Si, por ejemplo, un edificio se encuentra dañado (prendido fuego) la vista debe reflejarlo al usuario final. Lo mismo ocurriría si el edificio no posee suministro eléctrico o de agua.

**Opcionales ( para subir nota )**

* Se pueden incluir todo tipo de sonidos (incluyendo música de fondo) en la aplicación con la correspondiente funcionalidad de poder prenderlos y apagarlos a gusto del usuario del juego.
* Cuando se da la catástrofe terremoto, se puede incluir la animación que hace “vibrar” la pantalla para dar un mejor efecto.
* Se pueden agregar niveles de dificultad al juego (Fácil, Medio, difícil) comenzando con menos dinero en cada caso y aumentando las probabilidades de catástrofes.
* Efectos de transiciones (tipo ppt) entre los cambios de las interfaces gráficas.

# 

# **Entregables**

* Código fuente de la aplicación completa, incluyendo también: código de la pruebas, archivos de recursos
* Script para compilación y ejecución (ant)
* Informe, acorde a lo especificado en este documento

## Formas de entrega

## Habrá **4 entregas formales**. Las mismas tendrán una calificación de APROBADO o NO APROBADO en el momento de la entrega.

Aquél grupo que acumule 3 no aprobados, quedará automáticamente desaprobado con la consiguiente pérdida de regularidad en la materia. En cada entrega se debe traer el informe actualizado.

1er Entrega: *Mínimamente* (se aconseja avanzar lo más posible) pruebas funcionando que contemplen:

* Pruebas unitarias a los edificios.
* Pruebas de conexión de líneas de tensión sólo con centrales eléctricas y alimentando edificios.
* Pruebas de conexión de la red de agua.
* Pruebas de reparación de los bomberos.

2da Entrega: Modelo del Juego Completo con todas las pruebas unitarias y de integración que abarquen todos los casos del enunciado.

3er Entrega: Interfaz gráfica Parcial. A determinar por el ayudante.

4ta y última Entrega: Trabajo Práctico completo y funcionando con persistencia e interfaz gráfica final.

## Fechas de entrega programadas

1er Entrega: *Miércoles 12 / 11 / 2014 - Jueves 13 / 11 / 2014*

2da Entrega: *Miércoles 19 / 11 / 2014 - Jueves 20 / 11 / 2014*

3er Entrega: *Miércoles 26 / 11 / 2014 - Jueves 27 / 11 / 2014*

4ta y última Entrega: *Miércoles 03 / 12 / 2014 - Jueves 04 / 12 / 2014*

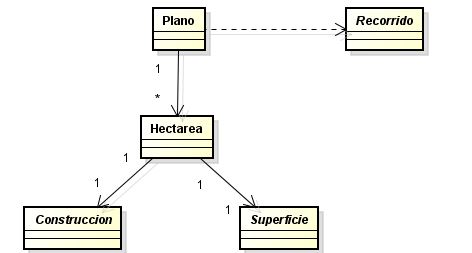
Informe

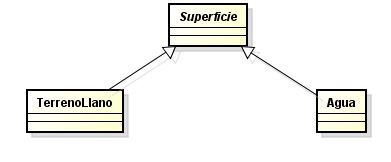
Supuestos

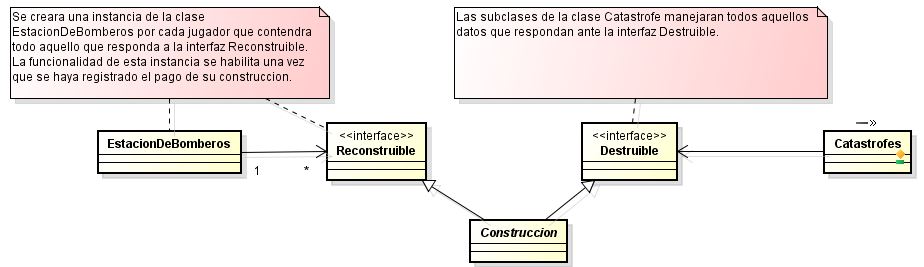
Se creara una instancia de la clase EstacionDeBomberos por cada jugador que contendrá todo aquello que responda a la interfaz Reconstruible. La funcionalidad de esta instancia se habilita una vez que se haya registrado el pago de su construcción y no ocupa ninguna hectárea.

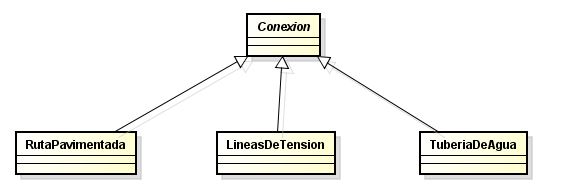
Las catástrofes pueden hacer un listado de las construcciones dañadas que luego se pasara a los bomberos. (En la realidad lo consideraríamos que en cada construcción dañada sus habitantes informan a los bomberos)

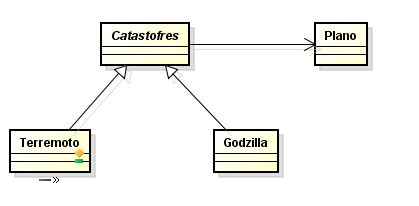
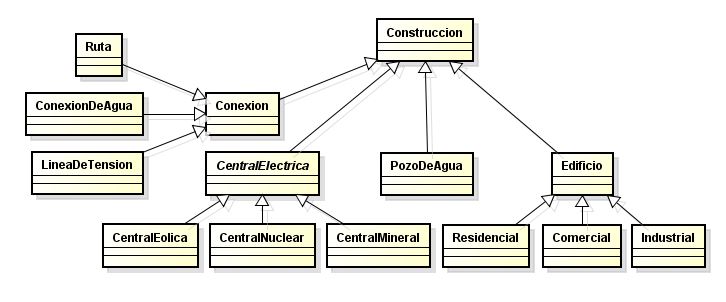
Diagramas de clases





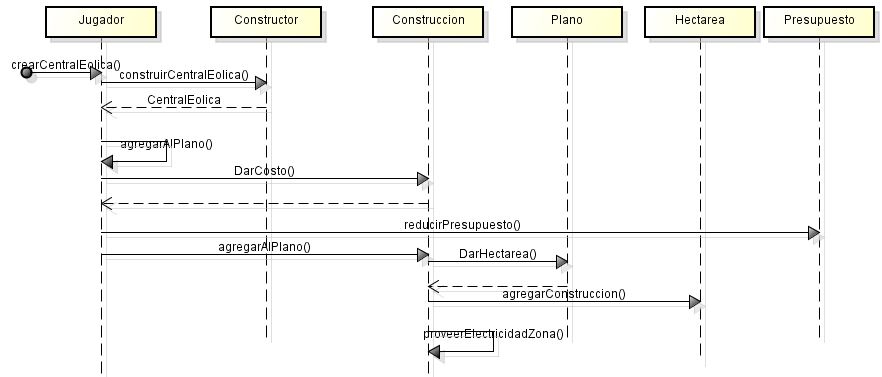




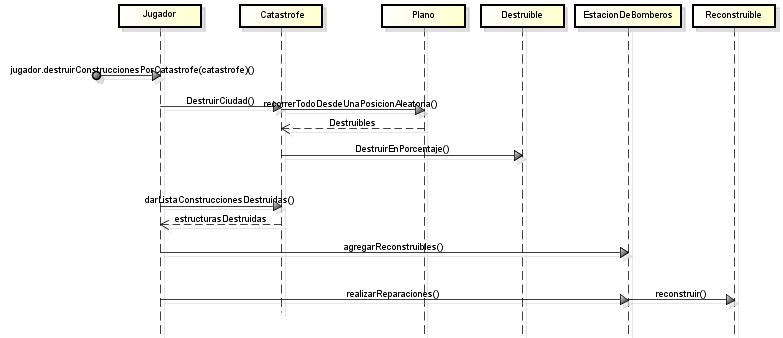


Diagramas de secuencia

(Construcción de una Central Eléctrica Eólica y su correspondiente distribución de electricidad a las zonas circundantes.)



(Catástrofe destruye construcciones de la ciudad y bomberos deben repararla)



Detalles de implementación

Uno de los puntos más conflictivos tenían que ver con la delegación de responsabilidades con respecto a los servicios y las construcciones, para resolver esto se intentó que las propiedades de los servicios (electricidad, agua y acceso al tránsito) los tenga la clase Hectárea (a través de la clase AdministradorServicios los controla) y la misma clase Hectárea es la que tiene o no la construcción que requiere dichos servicios.

El problema de la cantidad de construcciones distintas con requisitos de servicios distintos y propiedades distintas se solucionó simplemente usando lo que sabíamos de herencia y polimorfismo, separando a las construcciones residenciales, comerciales e industriales de las centrales eléctricas por ejemplo.

Otro punto conflictivo tenía que ver con las construcciones reparables que se le pasan a los bomberos. El problema principal radicaba en que cada vez que los bomberos debían reparar las construcciones tenían que hacer un barrido completo de todo el mapa, generando una demora excesiva en mapas medianos o grandes. Para corregir esto hicimos que las mismas catástrofes generen una lista de construcciones dañadas y el juego se encargue de pasarle esa lista a la estación de bomberos. Al acotar la lista de construcciones a una lista menor, el barrido tarda muchísimo menos tiempo.

Excepciones

Se utilizan 13 excepciones distintas:

1. ExcepcionCentralElectricaNoPoseeRedDeAgua.
2. ExcepcionCentralNoPuedeProveerMasElectricidad.
3. ExcepcionDineroInsuficiente.
4. ExcepcionElConsumoElectricoDeLaHectareaExcedeALaCapacidadMaxima.
5. ExcepcionHectareaNoBrindaLosServiciosNecesarios.
6. ExcepcionHectareaNoHabilitadaParaConstruirEdificio.
7. ExcepcionHectareaYaContieneUnaConstruccion.
8. ExcepcionJugadorIngresadoVacio.
9. ExcepcionJugadorYaExistente.
10. ExcepcionNoSePuedeConectarPorqueNoHayOtraLineaDeTensionAlrededor.
11. ExcepcionNoSePuedeConstruirEnEsteTerreno.
12. ExceptionEstacionDeBomberosNoEstaHabilitada.
13. ExceptionConstruccionComplemetamenteReparada.

Esta última es la única que no arroja un mensaje de error por pantalla, sino que saca la construcción de la lista de reparables que tenia la estación de bomberos. Las otras doce excepciones muestran un mensaje por pantalla indicando que la acción no se puede realizar (y una breve explicación de cuál fue la causa de que no se pueda realizar)

# 

# **Checklist de corrección**

Esta sección es para uso exclusivo de los docentes, por favor no modificar.

## Carpeta

**Generalidades**

* ¿Son correctos los supuestos y extensiones?
* ¿Es prolija la presentación? (hojas del mismo tamaño, numeradas y con tipografía uniforme)

**Modelo**

* ¿Está completo?¿Contempla la totalidad del problema?
* ¿Respeta encapsulamiento?
* ¿Hace un buen uso de excepciones?
* ¿Utiliza polimorfismo en las situaciones esperadas?

## Diagramas

**Diagrama de clases**

* ¿Está completo?
* ¿Está bien utilizada la notación?

**Diagramas de secuencia**

* ¿Está completo?
* ¿Es consistente con el diagrama de clases?
* ¿Está bien utilizada la notación?

**Diagrama de estados**

* ¿Está completo?
* ¿Está bien utilizada la notación?

# **Código**

**Generalidades**

* ¿Respeta estándares de codificación?
* ¿Está correctamente documentado?

1. <http://es.wikipedia.org/wiki/SimCity_(videojuego)> [↑](#footnote-ref-2)