

# UbikSim Web Service

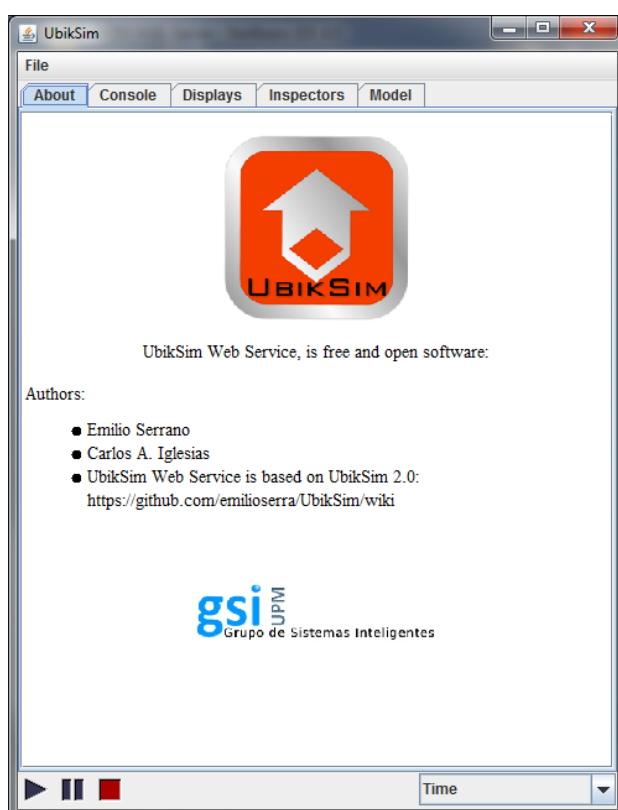
---

*Manual y notas de versión.*

*Emilio Serrano, emilioserra.com*

## Contenido

1	Instalación y entorno de trabajo .....	2
2	Instrucciones .....	2
2.1	Control.....	2
2.2	Información posicional .....	3
3	Principales elementos de la versión.....	3
4	Ejemplos de entrada/salida.....	4
5	Bugs y todo .....	6



## 1 Instalación y entorno de trabajo

- Idealmente, se debería desarrollar en base a las salidas del simulador y realizar peticiones para programar nuevas salidas (o tratar entradas dadas por otras aplicaciones).
  - La idea es facilitar migraciones a futuros simuladores.
- Distribuido como proyecto NetBeans de aplicación Web.
  - Si se pasa a otro IDE tipo Eclipse, recordad que hay clases repetidas en UbikSim2.jar y el proyecto (clases adaptadas). Así que es importante configurar el proyecto para leer primero ficheros propios y luego de librerías. Lo contrario resultaría en fallos.
- Librerías,
  - Por motivos de espacio, la mayoría están distribuidas [online](#).
  - Hay que enlazar a estas librerías desde el proyecto (Netbeans o eclipse si se prefiere).
  - Entre estas, hay que eliminar la referencia a OCPCore.jar, en su lugar, UbikSim web service se distribuye con el directorio lib que incluye OCPCoreV2.jar y librerías para JSON.
    - El motivo es que la versión inicial de OCPCore contiene la clase Servlet.class que entra en conflicto con TOMCAT.
- ETSITSimGUI tiene un par de rutas absolutas cableadas que hay que cambiar en cada máquina.
  - PATHBASE and ENVIRONMENTPATH.
  - Ver bugs y todo para solucionar esto.
- Si surgen más problemas en su instalación y ejecución, revisar lista de [problemas en instalación de UbikSim](#).
  - Normalmente los usuarios tiene problemas instalando Java 3D, a lo que se recomienda probar cualquier clase simple que use Java 3D en tú proyecto para comprobar que efectivamente es accesible en compilación y ejecución.

## 2 Instrucciones

- Una vez se ejecuta se ve la ventana de bienvenida. Se puede controlar la simulación accediendo a la dirección <http://localhost:8080/UbikSimMOSI-AGIL-Server/ubiksim> con varios parámetros descritos a continuación.
  - (cambiar localhost por IP).

### 2.1 Control

- Ejemplo: <http://localhost:8080/UbikSimMOSI-AGIL-Server/ubiksim?control=pause>
- Este parámetro interactúa con la consola MASON. *Pause*: monta simulación (primera pulsación); la deja correr (segunda), la pausa (tercera). *Play*: si ya está montada la simulación hace que avance un paso. *Stop*: para simulación, se requiere pause para empezar de nuevo.

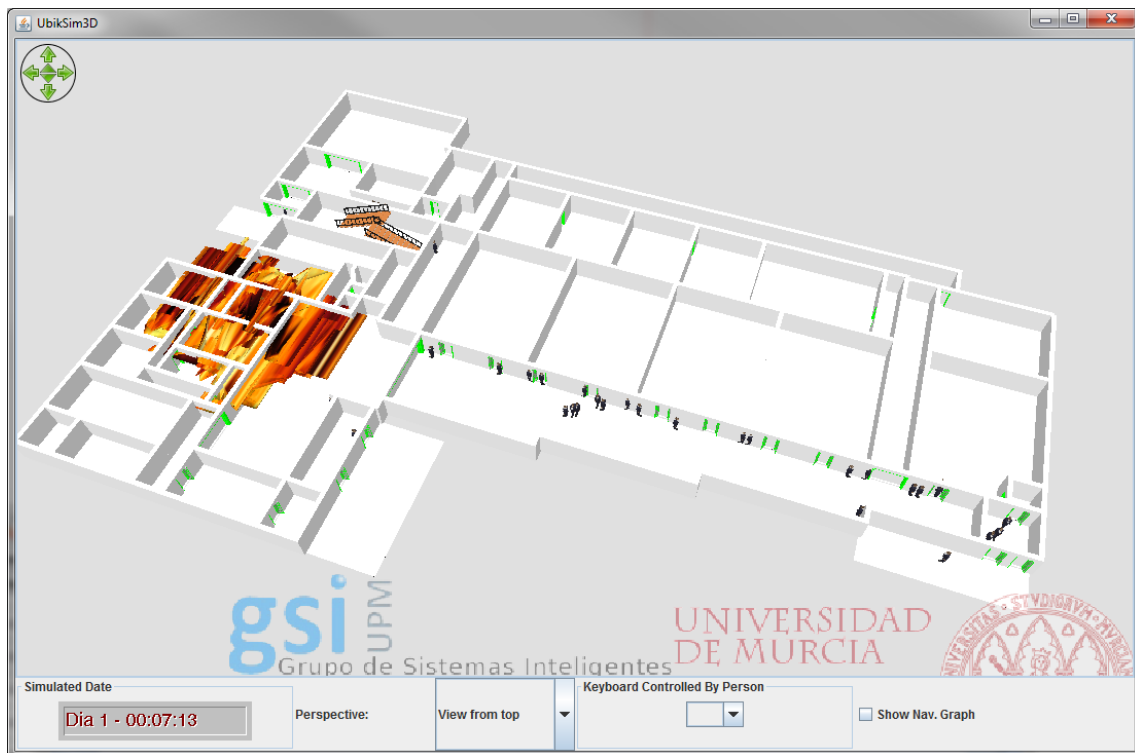
- Se añade el valor *control=frames* que muestra todos los frames del simulador en el lado del servidor (2D y 3D).
- Las acciones de control devuelven pequeños mensajes de confirmación con el step de la simulación. Esto se puede usar para regular la velocidad de manera remota.

## 2.2 Información posicional

- Ejemplo: <http://localhost:8080/UbikSimMOSI-AGIL-Server/ubiksim?position=people>
- position=people. Devuelve posición de entes simulados en un fichero JSON (identificador, habitación y coordenadas).
- position=map. Devuelve el mapa en triplas x, y, obstáculo. Donde obstáculo es 1 si hay una pared o puerta cerrada.
- position=emergency. Simula un fuego en posición aleatoria y devuelve la posición en fichero JSON (x,y y habitación).
- position=“(id,x,y)”. Tiple id, x, y. Se añade dicho usuario a la simulación si no se ha añadido ya y si esa posición está libre. Si se ha añadido ya el usuario, se cambia su posición a la pasada. Se devuelve un json con las rutas a las salidas. En el cálculo de las rutas considera entes simulados actuales que puedan bloquear caminos.

## 3 Principales elementos de la versión

- web\WEB-INF\environments. Contiene el plano de la simulación e información sobre este.
- web\index.html. Web de presentación y para operar.
- src\java\ubiksimdist
  - ETSITSim. Simulación sin GUI.
  - ETSITSimGUI. Simulación con GUI, llama a ETSITSim.
  - StartUbikEditor. Lanza editor para estudiar o modificar el entorno.
  - UbikSimServlet. Servlet que recoge peticiones web, básicamente de control o posición.
  - ServletToSimulator. Clase que coge los parámetros de las peticiones del servlet y los traduce en acciones a realizar en el simulador.



## 4 Ejemplos de entrada/salida

<http://localhost:8080/UbikSimMOSI-AGIL-Server/ubiksim?control=pause>

Simulation paused, time step -1

<http://localhost:8080/UbikSimMOSI-AGIL-Server/ubiksim?control=play>

Simulation played, time step 0

<http://localhost:8080/UbikSimMOSI-AGIL-Server/ubiksim?control=frames>

Simulation frames have been shown in the server side.

<http://localhost:8080/UbikSimMOSI-AGIL-Server/ubiksim?position=emergency>

Emergency in (99,172)

<http://localhost:8080/UbikSimMOSI-AGIL-Server/ubiksim?position=emergency>

There is already a fire set in (99,172)

[http://localhost:8080/UbikSimMOSI-AGIL-Server/ubiksim?position=\(Alicia,44,50\)](http://localhost:8080/UbikSimMOSI-AGIL-Server/ubiksim?position=(Alicia,44,50))

```

{
  "YARD1": [
    "(44,50)",
    "(44,51)",
    "(44,52)",
    "(44,53)",
    "(44,54)",
    "(44,55)",
    "(44,56)",
    ...
  ],
  "YARD3": [
    "(44,50)",
    "(44,51)",
    "(44,52)",
    "(44,53)",
    ...
  ],
  "YARD2": [
    "(44,50)",
    "(45,50)",
    ...
  ],
  "(272,134)",
  "(272,135)"
]
}

```

<http://localhost:8080/UbikSimMOSI-AGIL-Server/ubiksim?position=map>

```

{
  "height": 190,
  "obstacles": [
    "(0,0)",
    "(1,0)",
    "(2,0)",
    "(3,0)",
    ...
    "(292,189)",
    "(293,189)",
    "(294,189)"
  ],
  "width": 295
}

```

<http://localhost:8080/UbikSimMOSI-AGIL-Server/ubiksim?position=people>

```

{
  "a63": {
    "positionX": 121,
    "positionY": 168,
    "room": "YARD3"
  }
}

```

```

    },
    "a64": {
        "positionX": 158,
        "positionY": 66,
        "room": "051.0"
    },
    "a61": {
...
    },
    "Alicia": {
        "positionX": 44,
        "positionY": 50,
        "room": "046.1"
    },
...
    "a94": {
        "positionX": 52,
        "positionY": 109,
        "room": "038.0"
    }
}

```

## 5 Bugs y todo

- Usar rutas relativas en ETSITSimGUI. Podía obtenerlas desde servlet con el contexto (getServletContext()), pero tendría que pasar esa ruta a través de varias clases: Servlet-> ServletToSimulation ->ETSITSimGUI. Además se ejecutaría en cada petición o creación de servlet. Además, UbikSim usa muchas rutas relativas que cambian, así que exigiría un trabajo de depuración importante.
- Creo que casca si reinicias simulación en remoto (pause, stop, pause). Puede que los agentes worker no finalicen correctamente, no le he dado importancia.
- Para crear usuarios se podría usar el concepto de sesión del servlet de manera que te ahorres el parámetro id, pero me preocupa que por timeout se pierda control en simulaciones largas.
- El cálculo de caminos sigue siendo algo pesado. Se puede aumentar tamaño de celda o usar navmeshes. Se hace todavía más pesado si se usa Pathfinder en lugar de PathfinderThread.
- Map y people se puede usar para mostrar un mapa completo en el visor html o cualquier otra herramienta. También se puede generar otro mapa JSON en el formato que se desee.
- Se pueden añadir control de las personas simuladas: eliminar, añadir, etc.
- La emergencia no invalida caminos, se pueden ir marcando obstáculos en el mapa para que se automatice el cálculo (o, al igual que para conflictos con las personas, incluir la comprobación en siguiente paso).

