

1. Diseñe un ambiente de simulación en donde una entidad es la responsable de generar requerimientos de procesamiento (instancias de **WorkUnit_t**). Estas estructuras definen un requerimiento de procesamiento: consisten en un identificador del requerimiento, un puntero a la función que hay que ejecutar para servirlo, un contexto con información acerca de la tarea a realizar (parámetros) y una estructura con un conjunto de valores estadísticos que caracterizan la vida del requerimiento.

```
typedef void (*ProcFunc_t)(void *ctx);

typedef unsigned int WorkUnitId;

typedef struct {
    Time_t submitTime; // submission time
    Time_t startProcTime; // processing start time
    Time_t endProcTime; // processing end time
} WorkUnitStat_t;

typedef struct {
    WorkUnitId id;
    ProcFunc_t fun;
    void *context;
    WorkUnitStat_t stats;
} WorkUnit_t;
```

Estos requerimientos de procesamiento deben ser recibidos por un servidor de procesamiento (**WorkServer_t**) encargado de la distribución de los mismos entre distintos threads de procesamiento (**WorkerThread_t**).

Por último, diseñe un generador de requerimientos (**FakeWorkUnitGen_t**) encargado de generar requerimientos y ponerlos a disposición del work server para su procesamiento, y un monitor de estadísticas (**StatMonitor_t**) encargado de sacar estadísticas de funcionamiento.

Ensaye distintos esquemas:

- a. Cola única de requerimientos (El server coloca los requerimientos y los threads consumen de esa cola única).
- b. Múltiples colas de requerimientos (una por working thread) (El server coloca el requerimiento sobre alguna de las colas (criterios?), y el thread consume de su cola propia).