

## Módulo TQSEXEC

Os objetos do tipo `TQSExec` permitem a execução do gerenciador TQS e alguns de seus comandos, como o processamento global. É possível montar uma fila de edifícios a processar e processá-los forma automática.

### Modo de funcionamento

A execução de comandos no gerenciador é feita pelo objeto `TQSExec.Job`. Você entra tarefas tipo `TQSExec.Task` na fila, e no final o comando `Job.Execute` chama o gerenciador e executa os comandos.

### Definição de tarefas

As tarefas a serem realizadas são objetos derivados da classe `TQSExec.Task`. Cada objeto derivado define uma tarefa diferente. A tarefa é definida na construção do objeto.

### Pasta atual e a classe TaskFolder

Alguns comandos são realizados a partir de uma pasta determinada. Use este objeto para posicionar o gerenciador na pasta desejada.

```
TQSExec .TaskFolder (buildingName, folderType=FOLDER_FRAMES,  
folderName="", subFolder=SUBFOLDER_NONE)
```

Define a pasta atual do edifício

```
buildingName Nome do edifício  
folderType Pasta tipo FOLDER_XXX  
folderName Nome da pasta (string) para FOLDER_FLOORS  
subFolder Subpasta tipo SUBFOLDER_XXXX para FOLDER_FLOORS
```

As constantes que podem ser usadas em `folderType` são:

```
TaskFolder .FOLDER_FRAMES
```

Pasta Espacial

```
TaskFolder .FOLDER_COLUMNS
```

Pasta Pilares

```
TaskFolder .FOLDER_FOUNDATIONS
```

Pasta Fundações

```
TaskFolder .FOLDER_FLOORS
```

Pasta do pavimento

```
TaskFolder .FOLDER_GENERAL
```

Pasta Gerais

```
TaskFolder .FOLDER_INFRASTRUCTURE
```

Pasta Infraestrutura

As constantes que podem ser usadas em `subFolder` são:

```
TaskFolder .SUBFOLDER_NONE
```

Pasta do pavimento

```
TaskFolder .SUBFOLDER_BEAMS
```

Subpasta vigas, quando `folderName = TaskFolder.FOLDER_FLOORS`

```
TaskFolder .SUBFOLDER_STAIRS
```

Subpasta escadas, quando `folderName = TaskFolder.FOLDER_FLOORS`

## Pasta raiz do edifício

```
TQSExec .TaskRootFolder (rootfolder)
```

Define a pasta raiz de edifícios

## Processamento global

Esta é a principal tarefa definida, com processamento global de toda a estrutura. O preenchimento de dados é semelhante à janela de dados do gerenciador.

```
TQSExec .TaskGlobalProc (floorPlan=2, floorDraw=0, slabs=0,  
beams=1, columnsData=1, columns=0, columnsReport=1,  
gridModel=1, gridDraw=1, gridExtr=1, gridAnalysis=1,  
gridBeamsTrnsf=0, gridSlabsTrnsf=1, gridNonLinear=0,  
frameModel=1, frameAnalysis=1, frameBeamsTrnsf=1,  
frameColumnsTrnsf=1, foundations=0, stairs=0,  
fire=0, precastPhases=0)
```

Processamento global conforme parâmetros escolhidos

`FloorPlan` (0) Não (1) Extrair plantas (2) Extrair e processar

`floorDraw` (0) Não (1) Desenhar formas

`slabs` (0) Não (1) Esforços (2) Esforços e desenho

`beams` (0) Não (1) Esforços (2) Dimensionamento, detalh (3) E desenho

`columnsData` (0) Não (1) gravação de dados de pilares

`columns` (0) Não (1) Dimensionamento, detalhamento (2) E desenho

`columnsReport` (0) Não (1) Relatório geral de pilares

`gridModel` (0) Não (1) Geração do modelo

`gridDraw` (0) Não (1) Desenho de dados de grelha

`gridExtr` (0) Não (1) Extração do desenho

`gridAnalysis` (0) Não (1) Análise de esforços

`gridBeamsTrnsf` (0) Não (1) Transferência de esforços para vigas

`gridSlabsTrnsf` (0) Não (1) Transferência de esforços para lajes

`gridNonLinear` (0) Não (1) Análise não linear

`frameModel` (0) Não (1) Geração do modelo

`frameAnalysis` (0) Não (1) Análise de esforços

`frameBeamsTrnsf` (0) Não (1) Transferência de esforços para vigas

`frameColumnsTrnsf` (0) Não (1) Transferência de esforços para pilares

`foundations` (0) Não (1) Dimensionamento, detalhamento (2) E desenho

`stairs` (0) Não (1) Dimensionamento, detalhamento e desenho

`fire` (0) Não (1) Verificação à incêndio

`precastPhases` (0) Não (1) Pré-moldados: Todas as etapas construtivas

## Planta de cargas

Entre este objeto na lista de tarefas para gerar a planta de cargas

```
TQSExec. TaskLoadsReport()
```

Planta de cargas com as últimas combinações fornecidas

## Desenho de lajes

Além dos desenhos de flexão positiva e negativa, punção e cisalhamento, este comando permite também gerar as faixas de esforços correspondentes.

```
TQSExec .TaskSlabDrawings (rebartop=1, rebarbot=1, rebarpunch=1,  
rebarshear=1, forcestop=1, forcesbot=1, forcespunch=1, forcesshear=1)
```

Desenhos de lajes

`rebartop` (0) Não (1) Ferros positivos

`rebarbot` (0) Não (1) Ferros negativos

`rebarpunch` (0) Não (1) Ferros de punção

`rebarshear` (0) Não (1) Ferros de cisalhamento

`forcestop` (0) Não (1) Faixas positivos

`forcesbot` (0) Não (1) Faixas negativos

`forcespunch` (0) Não (1) Faixas de punção

`forcesshear` (0) Não (1) Faixas de cisalhamento

## Desenho de lajes treliçadas

Geração dos 4 desenhos típicos no pós-processamento de lajes treliçadas. Exige que as armaduras tenham sido previamente calculadas.

```
TQSExec .TaskLatticeSlabs (latticeList=1, latticeBeams=1,  
rebarList=1, fillingList=1)
```

Desenhos de lajes treliçadas

`latticeList` (0) Não (1) Tabela de vigotas treliçadas

`latticeBeams` (0) Não (1) Planta de fabricação de vigotas

`rebarList` (0) Não (1) Tabela de ferros complementares

`fillingList` (0) Não (1) Tabela de enchimentos

## Desenho de lajes mistas nervuradas

Defina primeiro a pasta antes de acionar o desenho de lajes mistas nervuradas.

```
TQSExec .TaskCompositeSlabs ()
```

Desenhos de lajes mistas nervuradas

## Tabela de ferros com todos os desenhos de armação

Este comando não existe diretamente no gerenciador. Ele cria uma planta na pasta `GERAIS` com todos os desenhos de armação de todas as pastas do edifício, e gera uma tabela de ferros única. Esta tabela é interessante para avaliação da quantidade total de aço no edifício e para comparações de quantitativos.

```
TQSExec .TaskRebarSchedule ()
```

Tabela de ferros com todos os desenhos de armadura do edifício

## Processamento de escadas

Caso o dimensionamento, detalhamento e desenho de escadas não tenha sido feito no processamento global, pode ser feito aqui. O processamento é de um único piso, que precisa ser definido antes na pasta de `ESCADAS`.

```
TQSExec .TaskStairs ()
```

Dimensionamento, detalhamento e desenho de escadas

## Desenho de formas

Gera os desenhos típicos de formas de um pavimento. A pasta atual precisa ser definida previamente, a menos dos cortes do edifício, que são sempre gerados na pasta especial.

```
TQSExec .TaskFloorPlanDrawings (floor=1, columns=1, slabs=1,  
slabsDims=1, beams=1, loads=1, sections=1)
```

Desenho de formas

`floor` (0) Não (1) Planta de formas  
`columns` (0) Não (1) Planta de pilares  
`slabs` (0) Não (1) Verificação de nós de lajes  
`slabsDims` (0) Não (1) Verificação de medidas de lajes  
`beams` (0) Não (1) Nós e cargas em vigas  
`loads` (0) Não (1) Cargas em lajes  
`sections` (0) Não (1) Cortes do edifício

## Desenhos de pré-moldados

Adicione esta tarefa para gerar os desenhos de formas de pré-moldados

```
TQSExec .TaskPreCastDrawings ()
```

Desenho de pré-moldados

## Transferência de esforços para vigas pré-moldadas

```
TQSExec .TaskPreCastBeams ()
```

Transferência de esforços para cálculo de vigas pré-moldadas

## Transferência de esforços para fundações pré-moldadas

```
TQSExec .TaskPreCastFoundations ()
```

Transferência de esforços para cálculo de fundações pré-moldadas

## Dimensionamento, detalhamento e desenho de pilares pré-moldados

```
TQSExec .TaskPreCastColumns ()
```

Dimensionamento, detalhamento e desenho de pilares pré-moldados

## Dimensionamento, detalhamento e desenho de consolos pré-moldados

```
TQSExec .TaskPreCastCorbels ()
```

Dimensionamento, detalhamento e desenho de consolos pré-moldados

## Quantitativos de pré-moldados

```
TQSExec .TaskPreCastMaterials ()
```

Quantitativos de pré-moldados

## Execução de tarefas – classe Job

A objetos da classe `Job()` acumulam tarefas, que são objetos derivados da classe `Task()`, e são executados. Cada execução carrega o gerenciador no ar, processa e depois sai do gerenciador. Você cria o objeto com:

```
Job = TQSExec.Job()
```

## Entrada de uma tarefa

As tarefas são colocadas em uma fila e executadas na mesma sequência de entrada.

```
job .EnterTask (task)
```

Acumula uma tarefa da classe `Task` para execução

## Execução das tarefas

Acione o comando `job.Execute()` para executar todas as tarefas.

```
job .Execute ()
```

Executa os comandos acumulados, chama o TQS e sai no final

## Programa de teste TSTEXEC.py para a classe TQSExec

Este programa usa como exemplo o edifício “`Mod-Padrão`” distribuído para teste do programa TQS. Se não estiver na sua árvore de edifícios, primeiro faça a instalação. Descompacte o modelo `MODPLA.TQS`, localizado na pasta `USUARIO\NGE\TESTE` da instalação TQS.

O programa duplica este edifício, alterando o fck (C25, C30, C35 e C40) e o método de cálculo (Modelo IV e VI), no total de 8 combinações. Cada edifício duplicado recebe um nome distinto, identificando fck e modelo estrutural.

O edifício é carregado através de um objeto `TQSBuild.Building`:

```
Building = TQSBuild.Building ()  
istat = building.file.Open (nomedi)
```

Depois, é duplicado com alterações no nome, concreto e modelo estrutural:

```
nomediteste = nomedi + "_" + concrete + "_" + modeldesc  
building.model.structuralModel = structuralModel  
building.materials.beamConcreteStrength = concrete
```

```
building.materials.columnConcreteStrength = concrete  
building.materials.foundationConcreteStrength= concrete  
building.file.SaveAs (nomediteste)
```

**Finalmente é feito o processamento global, incluindo uma tabela de ferros na pasta GERAIS:**

```
job = TQSExec.Job ()  
job.EnterTask(TQSExec.TaskFolder(nomediteste,  
TQSExec.TaskFolder.FOLDER_FRAMES))  
job.EnterTask(TQSExec.TaskGlobalProc(floorDraw=1,slabs=2,  
beams=3, columns=2, foundations=1, stairs=1))  
job.EnterTask (TQSExec.TaskRebarSchedule ())  
job.Execute ()
```