

## Separar é fácil, mas e lembrar de tudo depois?

Vimos que é uma boa ideia separarmos nosso código em pequenas responsabilidades para ajudar na manutenção e legibilidade do nosso código. Porém, será que conseguimos juntar tudo depois para que nossa aplicação continue funcionando?

Não é raro esquecermos uma coisa aqui ou lá, será que há uma maneira de automatizarmos pelo menos alguma parte desse processo? Sim, há e podemos fazer isso através do módulo `consign`, que pode ser baixado pelo `npm`. Este módulo pode carregar automaticamente para nós todos os módulos dentro de alguma pasta em nosso sistema, inclusive passa **um** parâmetro para eles.

### Questão

Qual das opções abaixo usa o módulo `consign` em `alurapic/config/express` corretamente, para carregar todos os módulos da pasta `alurapic/app/api` e `alurapic/app/routes`. Lembre-se que `alurapic/app/routes` faz uso de `alurapic/app/api` e não o contrário.

Selecione uma alternativa

**A**

```
// alurapic/config/express.js

var express = require('express');
var consign = require('consign');
var app = express();

app.use(express.static('./public'));

consign({cwd: 'app'})
  .include('routes')
  .then('api');
  .into(app);

module.exports = app;
```

**B**

```
// alurapic/config/express.js

var express = require('express');
var consign = require('consign');
var app = express();

app.use(express.static('./public'));

consign({cwd: 'app'})
  .include('api')
  .then('routes')

module.exports = app;
```

**C**

```
// alurapic/config/express.js

var express = require('express');
var consign = require('consign');
var app = express();
```

```
app.use(express.static('./public'));

consign({cwd: 'app'})
  .include('api')
  .then('routes')
  .into(app);

module.exports = app;
```