

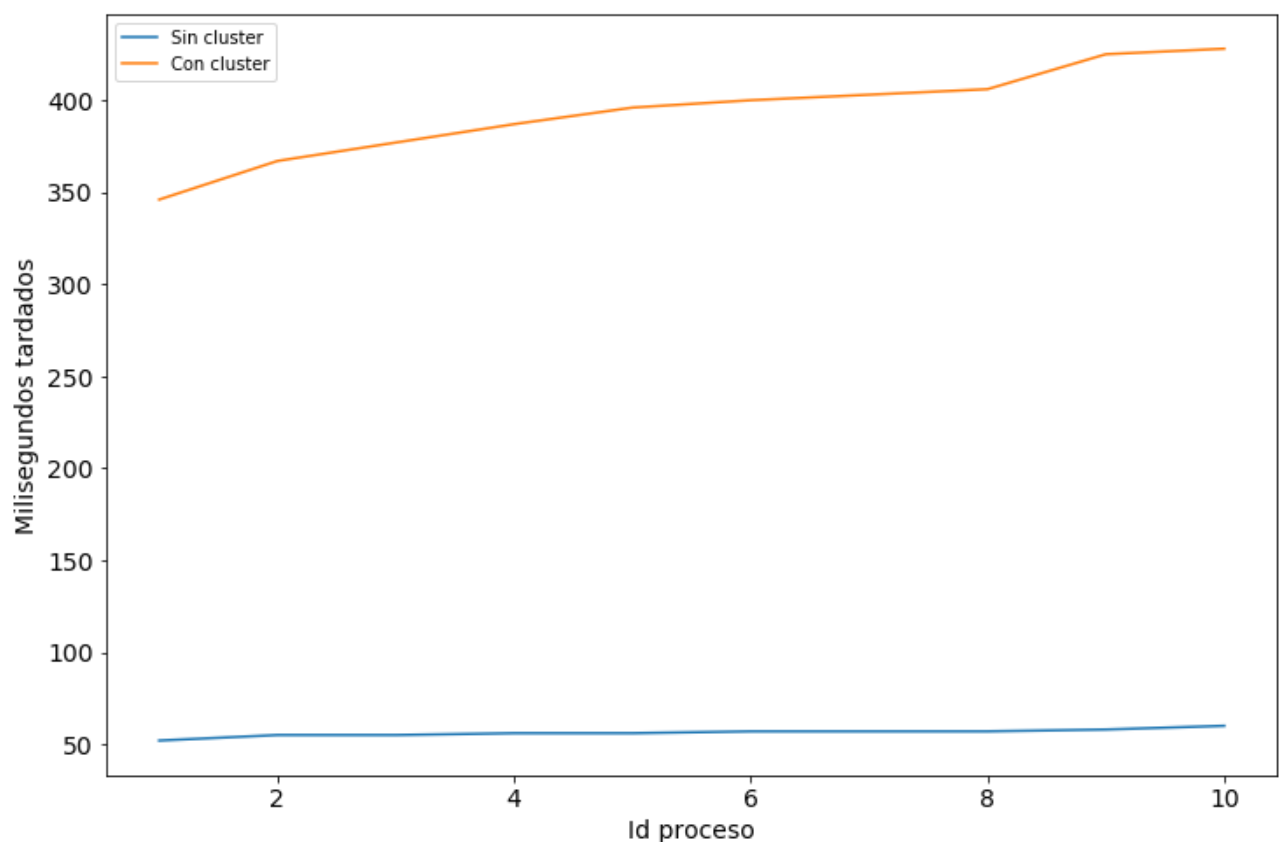
Nosotros usamos los scripts hechos en JS.

```
1 const axios = require('axios').default;
2 let promesas = [];
3 let segundosInicio;
4 for(let i = 0; i < 10; i++){
5     segundosInicio=new Date();
6     promesas.push(
7         axios.get('http://localhost:3000/funciones').then(res => {console.log("Tardo ",i,new Date()-segundosInicio)}))
8     );
9 }
10
11 Promise.all(promesas).then(res => {console.log("termine")});
```

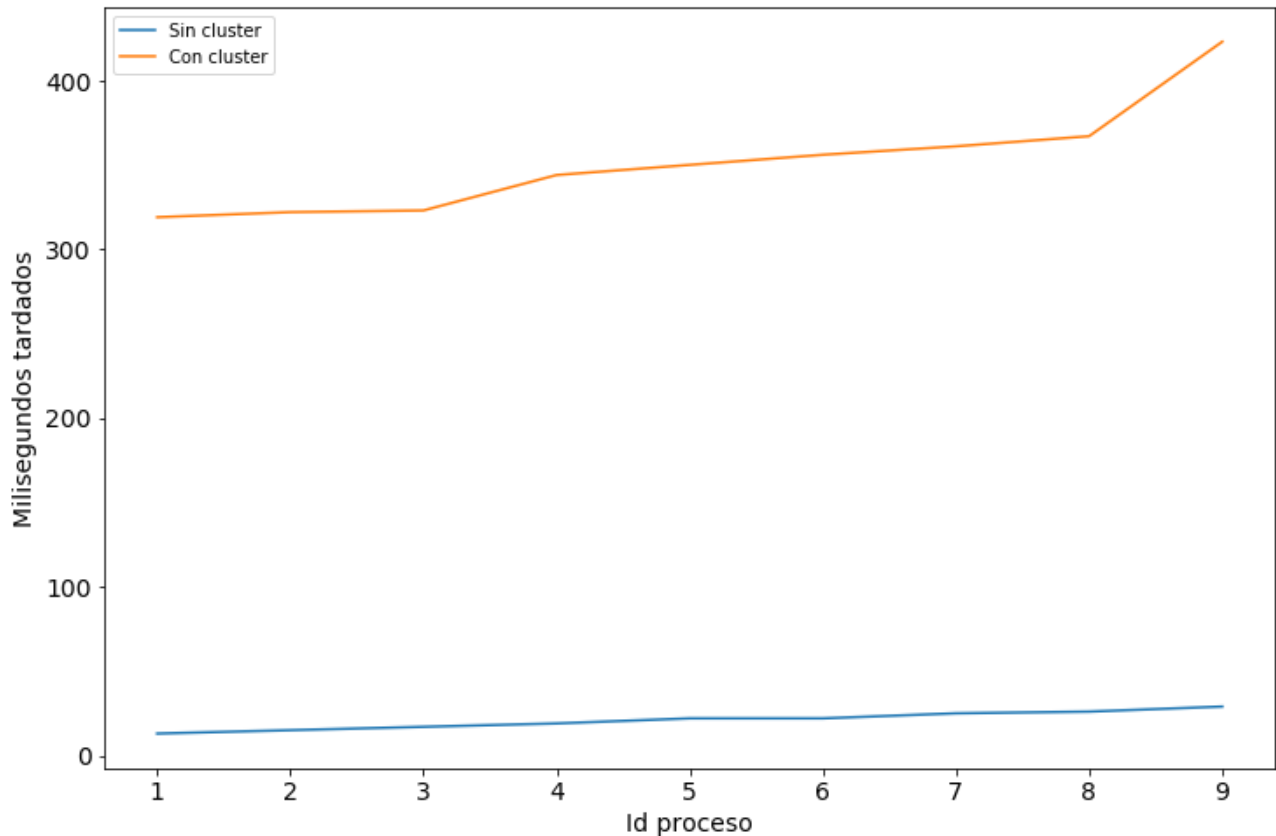
```
1 const axios = require('axios').default;
2 let promesas = [];
3 let segundosInicio = new Date();
4 promesas.push(
5     axios.post('http://localhost:3000/4/reservar', {usuario: '1',butacas: ["b3"]}).then(res => {console.log("Tardo 1",new Date()-segundosInicio)})
6     axios.post('http://localhost:3000/4/reservar', {usuario: '2',butacas: ["b4"]}).then(res => {console.log("Tardo 2",new Date()-segundosInicio)})
7     axios.post('http://localhost:3000/4/reservar', {usuario: '3',butacas: ["c2"]}).then(res => {console.log("Tardo 3",new Date()-segundosInicio)})
8     axios.post('http://localhost:3000/4/reservar', {usuario: '4',butacas: ["c3"]}).then(res => {console.log("Tardo 4",new Date()-segundosInicio)})
9     axios.post('http://localhost:3000/4/reservar', {usuario: '5',butacas: ["c4"]}).then(res => {console.log("Tardo 5",new Date()-segundosInicio)})
10    axios.post('http://localhost:3000/4/reservar', {usuario: '6',butacas: ["d1"]}).then(res => {console.log("Tardo 6",new Date()-segundosInicio)})
11    axios.post('http://localhost:3000/4/reservar', {usuario: '7',butacas: ["d2"]}).then(res => {console.log("Tardo 7",new Date()-segundosInicio)})
12    axios.post('http://localhost:3000/4/reservar', {usuario: '8',butacas: ["d3"]}).then(res => {console.log("Tardo 8",new Date()-segundosInicio)})
13    axios.post('http://localhost:3000/4/reservar', {usuario: '9',butacas: ["f1"]}).then(res => {console.log("Tardo 9",new Date()-segundosInicio)})
14 );
15
16 Promise.all(promesas).then(res => {console.log("termine")});
```

Cuando hicimos las pruebas con los scripts hechos en JS nos dieron estos valores

Funciones:



Reservar:



Los valores con las pruebas fueron raros porque con la versión clusterizada nos tardaba mas pero lograba terminar todas las transacciones correctamente y en la base se cambiaba todo. Pero cuando usamos la versión no clusterizada nos tarda menos pero en ninguno de los dos casos se termina de ejecutar el script y en la parte de reservar no se hacen correctamente los cambios en la base.

Después probamos con mandar 50 requests con el script y el cluterizado terminaba todo masomenos en 1300ms el no clusterizado ni terminaba, hacia 9 requests y luego tiraba que había muchas peticiones.