



nombre, tipo,
cupo,
edadMinima =
0

this.nombre = nombre;
this.tipo = tipo;
this.cupo = cupo;
this.edadMinima =
edadMinima;
this.inscritos = [];

usuario.actividadesInscritas.length >=
3

No puede inscribirse en
más de 3 actividades.

usuario.edad < this.edadMinima

No cumple con la edad
mínima

```
graph TD; Entry(( )) --> D1{this.cupo <= this.inscritos.length}; D1 --> B1[Actividad llena]; B1 --> D2{usuario.tipo === 'visitante' && this.tipo !==}; D2 --> B2[Los visitantes solo pueden asistir a charlas.]; D2 --> D3{this.inscritos.push(usuario); usuario.actividadesInscritas.push(this);}; D3 --> Exit(( ))
```

`this.cupo <= this.inscritos.length`

Actividad llena

`usuario.tipo === 'visitante' &&
this.tipo !==`

Los visitantes solo
pueden asistir a
charlas.

`this.inscritos.push(usuario);
usuario.actividadesInscritas.push(this);`

Inscripción
exitosa.

Sistema

Constructor

```
this.usuarios = [];  
this.actividades = [];  
this.noInscritosPorCupo  
= [];
```

usuario.edad < 16
&& !usuario.tutor

Usuario requiere
tutor.

procesarInscripciones

const usuario of
this.usuarios

actividadesCandidatas
=
this.actividades.filter

if
(usuario.tipo
=== 'visitante') return
act.tipo === 'charla';
if (usuario.tipo === 'docente') return true;
if (usuario.tipo === 'estudiante')
return usuario.edad >= 16;

True

if
(usuario.tipo
=== 'docente') {
const resultado =
actividadesCandidatas[0]?.inscribir(usuario);
if (!resultado?.exito) {

this.noInscritosPorCupo.push(usuario);
}
}

const actividad of
actividadesCandidatas

if
(usuario.actividadesInscritas.length
>= 3) break;

```
resultado =  
actividad.inscribir(usuario);
```

```
if (!resultado.exito && resultado.mensaje === 'Actividad llena.') {  
  this.noInscritosPorCupo.push(usuario);  
}
```

```
reporteFinal() {  
  const asistentesPorPerfil =  
    {};  
  const demandaPorActividad  
    = {};  
  const edadesPorActividad =  
    {};  
  let totalUsuariosCompleto =  
    0;
```

```
for (const usuario of this.usuarios) {  
  asistentesPorPerfil[usuario.tipo] =  
    (asistentesPorPerfil[usuario.tipo] || 0) + 1;  
  if (usuario.actividadesInscritas.length === 3) totalUsuariosCompleto++;  
  for (const act of usuario.actividadesInscritas) {  
    demandaPorActividad[act.nombre] = (demandaPorActividad[act.nombre] || 0) + 1;  
    if (!edadesPorActividad[act.nombre]) edadesPorActividad[act.nombre] = [];  
    edadesPorActividad[act.nombre].push(usuario.edad);  
  }  
}
```

```
const edadesPromedio = {};  
for (const [nombre, edades] of  
Object.entries(edadesPorActividad)) {  
  const sum = edades.reduce((a, b) =>  
    a + b, 0);  
  edadesPromedio[nombre] = (sum /  
    edades.length).toFixed(1);  
}
```

```
return {  
  asistentesPorPerfil,  
  actividadesMasDemandadas:  
    Object.entries(demandaPorActividad)  
      .sort((a, b) => b[1] - a[1]),  
  edadesPromedio,  
  porcentajeCompletaron:  
    ((totalUsuariosCompletos / this.usuarios.length)  
      * 100).toFixed(2) + '%'  
};
```

```
listarParticipantesPorActividad(nombreActividad) {  
  const actividad = this.actividades.find(a =>  
    a.nombre === nombreActividad);  
  return actividad ? actividad.inscritos.map(u =>  
    u.nombre) : [];  
}  
  
listarAsistentesPorTipo(tipo) {  
  return this.usuarios.filter(u => u.tipo === tipo);  
}
```




```
const sistema = new Sistema();
```



```
sistema.agregarActividad(new Actividad('Taller de Robótica Avanzada', 'taller', 10, 18));
sistema.agregarActividad(new Actividad('Charla de Innovación', 'charla', 20));
sistema.agregarActividad(new Actividad('Taller de Programación Básica', 'taller', 15, 16));

sistema.registrarUsuario(new Usuario('Ana', 20, 'estudiante', true));
sistema.registrarUsuario(new Usuario('Luis', 17, 'estudiante', true));
sistema.registrarUsuario(new Usuario('Carlos', 35, 'docente', true));
sistema.registrarUsuario(new Usuario('Marta', 30, 'profesional', true));
sistema.registrarUsuario(new Usuario('Tomás', 15, 'estudiante', true, 'Padre'));
sistema.registrarUsuario(new Usuario('Lucía', 25, 'visitante', true));
```



```
sistema.procesarInscripciones();
```



```
console.log(sistema.reporteFinal());
console.log('Participantes en Charla de Innovación:',
sistema.listarParticipantesPorActividad('Charla de Innovación'));
console.log('No inscritos por falta de cupo:',
sistema.noInscritosPorCupo.map(u => u.nombre));
```