

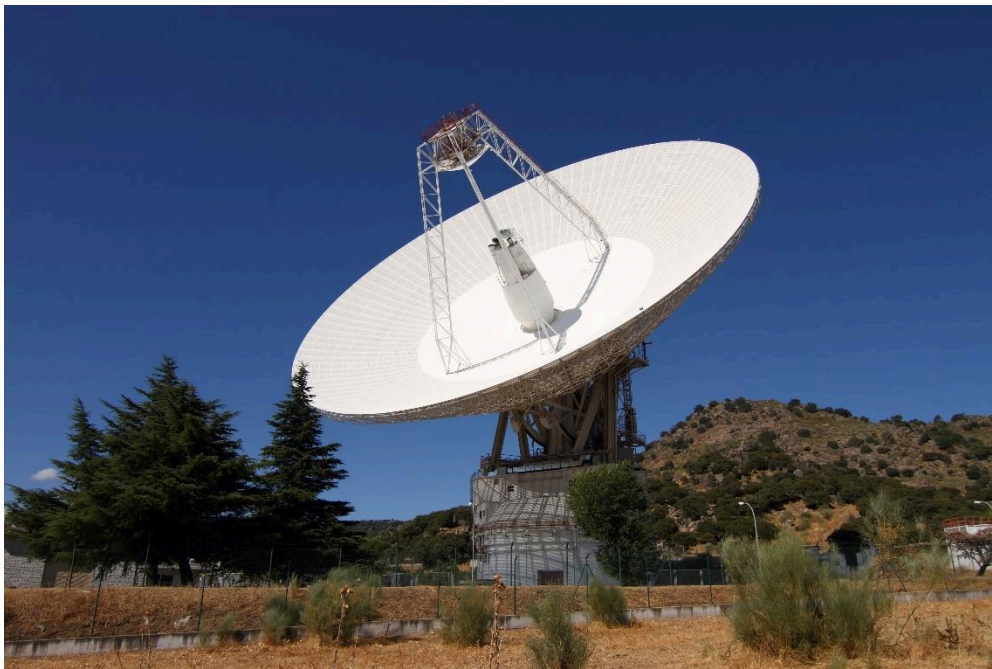


IIC3103 - Taller de Integración

Departamento Ciencia de la Computación
Escuela de Ingeniería
Pontificia Universidad Católica de Chile

Pontificia Universidad Católica

Tarea 2: *“Houston, we have a problem”*



Antena del [Deep Space Network](#) (DSN) de la NASA en Madrid, España. [CC-BY-SA](#).

Se han caído los sistemas de comunicación satelital de la NASA. Los satélites en órbita ocupan *WebSockets* para comunicarse con *Ground Control*. Es por ello que te han llamado a ti, una persona experta en dichas tecnologías, para poder reconstruir el comando de control de *Houston* y volver a tener control sobre los satélites que actualmente orbitan la Tierra.



IIC3103 - Taller de Integración

Departamento Ciencia de la Computación
Escuela de Ingeniería
Pontificia Universidad Católica de Chile

Pontificia Universidad Católica

Índice

Objetivo	3
Trabajo a realizar	3
WebSocket	3
Entregables	4
Penalizaciones	4
Versionamiento de código	4
Descripción general de la tarea	5
Componentes mínimos de la visualización	6
Resumen de Eventos	8
Bonus	9
Versiones y control de cambios	10
Anexo 1: Uso de Socket.io	11
Anexo 2: Librerías recomendadas	12
Anexo 3: Detalle de eventos	14
Anexo 4: Detalle de objetos	17



IIC3103 - Taller de Integración

Departamento Ciencia de la Computación
Escuela de Ingeniería
Pontificia Universidad Católica de Chile

Pontificia Universidad Católica

Objetivo

El objetivo de esta tarea es replicar un sistema con comunicación basada en eventos. El trabajo a realizar consistirá en enviar y recibir eventos en tiempo real utilizando *WebSockets* como capa de comunicación. Adicionalmente, se deberá construir una capa de visualización que permita interactuar con el sistema. Toda la tarea deberá ser implementada ocupando algún *framework* web. Algunos de estos *frameworks* son *React*, *Vue*, *Gatsby*, *Angular*, *Svelte*, o similar. También es posible utilizar *HTML*, *CSS* y *Javascript* puros (sin *framework*).

Trabajo a realizar

Deberán conectarse a un *WebSocket* que entregará eventos sobre las posiciones de satélites, los cuales se van actualizando en tiempo real.

Cada uno deberá construir un sitio web que permita desplegar esta información en un globo, que muestre todos los satélites, los cuales se deben ir actualizando en la medida que se van recibiendo eventos. Adicionalmente, se debe implementar un sistema de comunicación, que permita enviar y recibir mensajes de texto hacia los satélites.

En esta tarea no se debe guardar información en bases de datos, solo se debe mostrar la información que se empezó a consumir desde que se inició la conexión con el *WebSocket*.

WebSocket

Para esta entrega se ocupará únicamente la información entregada por el *WebSocket*. Para conectarse a este se ocupan los siguientes datos:

Conexión con *WebSocket*:

- **Protocolo:** wss://
- **Servidor:** tarea-2.2025-1.tallerdeintegracion.cl
- **Ruta:** /connect



IIC3103 - Taller de Integración

Departamento Ciencia de la Computación
Escuela de Ingeniería
Pontificia Universidad Católica de Chile

Pontificia Universidad Católica

Entregables

Para esta tarea, deberán entregar el link a su tarea desplegada, disponible públicamente en internet, donde se pueda entrar a probar las funcionalidades pedidas en la tarea. Para esto, se recomienda usar la capa gratuita de servicios tipo [Netlify](#), [Vercel](#), [Github Pages](#), entre otros.

Fecha de entrega

La tarea debe entregarse antes del 16 de abril antes de las 18:00, y deberá estar desplegada y disponible durante el periodo de corrección (2 semanas).

Requisitos mínimos

Las tareas que no cumplan con las siguientes condiciones no serán corregidas y serán evaluados con la nota mínima:

- **La página web deberá ser pública**, accesible desde cualquier dispositivo conectado a internet.
- **El código deberá estar versionado en su totalidad** en un repositorio Git.
- **El sitio implementado debe corresponder al código entregado**. Para la revisión, tareas serán corridas localmente por los ayudantes para comprobar el cumplimiento de este punto. En caso de que el código entregado no represente fielmente al sitio entregado, se calificará la tarea con la nota mínima.

Penalizaciones

- Se descontarán 0,2 puntos de la nota de la tarea por cada hora de atraso en la entrega, contados a partir de la fecha estipulada en el punto anterior.
- En caso de no poder desplegar la API, se evaluará con nota máxima 5,0.
- Cualquier intento de copia, plagio o acto deshonesto en el desarrollo de la tarea, será penalizado con nota 1,1 de acuerdo a la política de integridad académica del DCC.

Versionamiento de código

El sitio y todo su código fuente deberá estar versionado en un repositorio en GitHub, creado en el *classroom* del curso:

- Link para creación de repositorio: <https://classroom.github.com/a/AO3BJeMR>



IIC3103 - Taller de Integración

Departamento Ciencia de la Computación
Escuela de Ingeniería
Pontificia Universidad Católica de Chile

Pontificia Universidad Católica

Descripción general de la tarea

Se espera una página web que contenga un globo terráqueo que muestre los distintos satélites e información relacionada, entregada por el *WebSocket*. Asimismo, la página deberá contar con un chat para recibir información y entregar comandos; y una tabla con información adicional acerca de los satélites mostrados.

Si bien no se espera el uso de recursos de diseño gráfico y/o ilustración, si se espera que el sitio implementado tenga elementos básicos de usabilidad que permitan su uso y navegación.

El mockup a continuación es una interfaz de referencia. Recuerde incluir todos los elementos de la sección “componentes mínimos de la visualización”.

Tarea 2: Houston we have a problem

Satélites en órbita

Filtrar por país: Todos Filtrar por misión:

ID	Bandera	Satélite	Misión	Tipo	Potencia	Altitud	Vida útil
THU-157		Thuraya THU-157	Communication constellation for satellite telephony in EMEA and Asia	COM	23	36000.0	142
THU-158		Thuraya THU-158	Communication constellation for satellite telephony in EMEA and Asia	COM	40	36000.0	634
GAL-345		Galileo GAL-345	European navigation constellation	NAV	36	23222.0	48
GAL-346		Galileo GAL-346	European navigation constellation	NAV	36	23222.0	286
GAL-347		Galileo GAL-347	European navigation constellation	NAV	34	23222.0	769
GAL-348		Galileo GAL-348	European navigation constellation	NAV	25	23222.0	805
GAL-349		Galileo GAL-349	European navigation constellation	NAV	30	23222.0	920
GAL-350		Galileo GAL-350	European navigation constellation	NAV	37	23222.0	1038
BDS-338		BeiDou BDS-338	Chinese navigation constellation	NAV	47	21150.0	0
BDS-339		BeiDou BDS-339	Chinese navigation constellation	NAV	31	21150.0	16
BDS-		BeiDou BDS-340	Chinese navigation	NAV	45	21150.0	249

Chat con Satélites

Sistema (N/A) [13:10:14]: [AUTH] {"message": "AUTH OK. Operador (GGJJ)", "type": "AUTH"}

Escribe un mensaje...

Mockup de una versión de la tarea. Creación propia.



IIC3103 - Taller de Integración

Departamento Ciencia de la Computación
Escuela de Ingeniería
Pontificia Universidad Católica de Chile

Pontificia Universidad Católica

Componentes mínimos de la visualización

En el globo terráqueo

Ubicaciones terrestres

Antenas del Deep Space Network

- Goldstone Deep Space Communications Complex (35°25'36"N, 116°53'24"W)
- Madrid Deep Space Communications Complex (40°25'53"N, 4°14'53"W)
- Canberra Deep Space Communication Complex (35°24'05"S, 148°58'54"E)

Sitios de lanzamiento activos entregados por el WebSocket (Evento LAUNCHSITES)

Zonas de cobertura satelital

- Cada satélite cubrirá una zona circular sobre la superficie terrestre.
- Esta zona estará centrada en la posición del satélite.
- El radio de dicha zona será equivalente a la potencia del satélite en kilómetros.
- Deberá haber un botón que permita mostrar/ocultar estas zonas de cobertura.

Cálculo de cobertura para las antenas

- Al hacer click en cualquier antena, se debe mostrar un cuadro informativo incluyendo:
 - Nombre y posición de la antena.
 - Satélites cercanos (el punto está dentro de su zona de cobertura)
 - Señal recibida. Se debe calcular de la siguiente forma:
 - Sumar la señal para cada satélite cercano:

Máximo entre 0% y $(1 - \frac{\text{la distancia del punto geográfico al centro de la cobertura del satélite}}{\text{potencia del satélite}})$ dividido por la potencia del satélite.

Es decir, si el punto seleccionado es el centro de la cobertura del satélite (las coordenadas del satélite, en la superficie terrestre), la señal será 100%.

En los límites de la zona de cobertura y fuera de la zona de cobertura, la señal será 0% (el satélite no está cercano).

Para cualquier punto intermedio, la señal será proporcional a la distancia al centro de cobertura, de acuerdo a la fórmula.



IIC3103 - Taller de Integración

Departamento Ciencia de la Computación
Escuela de Ingeniería
Pontificia Universidad Católica de Chile

Pontificia Universidad Católica

Ubicaciones en el espacio

Arco de lanzamiento de satélites (Evento LAUNCH)

- Se deberá mostrar una línea que vaya desde el punto de lanzamiento hasta la debris_zone.
- Este arco se deberá mostrar mientras el satélite esté en camino a órbita.
- Este arco deberá desaparecer una vez que el satélite concluya su lanzamiento

Posición actual de los satélites

- Se deberá indicar cada satélite como un punto en el espacio.
- Cada tipo de satélite deberá tener un color distinto



IIC3103 - Taller de Integración

Departamento Ciencia de la Computación
Escuela de Ingeniería
Pontificia Universidad Católica de Chile

Pontificia Universidad Católica

Tablero informativo

Debe mostrar todos los satélites en órbita y los siguientes atributos para cada satélite:

- id
- Bandera del País y organización (ej. "🇺🇸 NASA"). *Sugerencia:* ocupar [FlagCDN](#) o [FlagsAPI](#).
- Nombre de la misión
- Nombre del satélite
- Fecha de lanzamiento
- Tipo
- Potencia
- Vida útil

Los satélites deben estar ordenados por altitud.

La tabla debe poder filtrar por país y misión.

Chat

Debe tener un panel que muestre todos los mensajes entrantes, así como los salientes, en una interfaz gráfica tipo chat, identificando fecha y hora de los mensajes, así como el emisor (`station_id/satellite_id`) de estos.

Debe poseer un cuadro de texto que permita escribir y enviar un nuevo mensaje.

Debe mostrar mensajes con level "warn" en color rojo.

Información adicional

Al hacer click en un satélite, debe mostrar toda la información relevante sobre el satélite.

Al hacer click en un sitio de lanzamiento, debe mostrar toda la información relevante sobre el sitio

Realce visual con la ubicación de todos los eventos **LAUNCH**.

Realce visual (con duración cercana a 20 segundos) en la última posición registrada un satélite que sufrió **CATASTROPHIC-FAILURE**.

Las notificaciones o alertas deben mostrarse durante un tiempo acotado para no colapsar el mapa con demasiados elementos.



IIC3103 - Taller de Integración

Departamento Ciencia de la Computación
Escuela de Ingeniería
Pontificia Universidad Católica de Chile

Pontificia Universidad Católica

Resumen de Eventos¹

Emitidos por el WebSocket

CATASTROPHIC-FAILURE: Indica que un satélite es destruido. Se desconoce el error.

COMM: Permite al operador recibir mensajes vía chat.

DEORBITING: Indica que un satélite cumplió su vida útil y se deja estrellar con la tierra. Es destruido.

IN-ORBIT: Indica que un satélite alcanzo su órbita.

LAUNCH: Indica que se lanzó un satélite desde un sitio de lanzamiento.

LAUNCHSITES: Muestra una lista de los sitios de lanzamiento.

POWER-DOWN: Indica que se disminuyó la potencia del satélite. Disminuye su cobertura.

POWER-UP: Indica que se aumentó la potencia del satélite. Aumenta su cobertura.

REQUEST: Solicita una petición al operador.

SATELLITES: Muestra una lista de los satélites en órbita y los lanzados y en camino a órbita.

SATELLITE-STATUS: Muestra información del satélite.

Recibidos por el WebSocket

AUTH: Permite al operador acceder al sistema y recibir los comandos del WebSocket.

COMM: Permite al operador mandar mensajes vía chat.

LAUNCHSITES: Permite solicitar una lista de los sitios de lanzamiento.

SATELLITES: Permite solicitar una lista de los satélites en órbita y los lanzados y en camino a órbita.

SATELLITE-STATUS: Permite solicitar el status de un satélite particular.

Comandos

¹ El detalle de los eventos está disponible en el Anexo 3: *Detalle de Eventos*



IIC3103 - Taller de Integración

Departamento Ciencia de la Computación
Escuela de Ingeniería
Pontificia Universidad Católica de Chile

Pontificia Universidad Católica

Además de mensajes normales, en los eventos tipo **COMM** podrás enviar comandos para interactuar con los satélites. Los comandos siempre parten con / (slash).

/destroy <satellite_id>: Permite al operador destruir a distancia un satélite.

/power <satellite_id> <power>: Permite cambiar la potencia de un satélite.

Bonus

(+0,5 puntos adicionales)

Se asignarán 0,5 puntos adicionales a la nota a quienes implementen en su visualización la ubicación en tiempo real de la Estación Espacial Internacional (ISS).

La representación de la *ISS* deberá ser evidente para el usuario de la página, recomendándose el uso de *labels*, colores u otros tipos de mensajes con tal de diferenciar la *ISS* de otros satélites.

Solo se considerarán las implementaciones que ocupen la *API* de la NASA: <https://api.nasa.gov/>

Una forma de comprobación será verificando la ubicación con la obtenida a través del *Live Space Station Tracking Map*, disponible en el siguiente link: https://spotthestation.nasa.gov/tracking_map.cfm



IIC3103 - Taller de Integración

Departamento Ciencia de la Computación
Escuela de Ingeniería
Pontificia Universidad Católica de Chile

Pontificia Universidad Católica

Versiones y control de cambios

(En caso de haber actualizaciones o correcciones se anotarán aquí)

0.1.0	25/03/2025	Primer borrador de la tarea
1.0.0	31/03/2025	Publicación tarea 2



IIC3103 - Taller de Integración

Departamento Ciencia de la Computación
Escuela de Ingeniería
Pontificia Universidad Católica de Chile

Pontificia Universidad Católica

Anexo 1: Uso de Socket.io

Para esta tarea deberán investigar cómo conectarse a un *WebSocket* en el *framework* de su elección. Ahora, como equipo docente **les recomendamos fuertemente no utilizar la librería *Socket.io***, dado como lo indica su [documentación](#), ***Socket.io* NO es una implementación de *WebSocket***.

Este paquete, si bien simplifica algunos aspectos de *WebSocket*, solo puede comunicarse con *WebSocket* hechos en *Socket.io*, lo cual no es el caso para el servidor de la tarea.



IIC3103 - Taller de Integración

Departamento Ciencia de la Computación
Escuela de Ingeniería
Pontificia Universidad Católica de Chile

Pontificia Universidad Católica

Anexo 2: Librerías recomendadas

Para el desarrollo de esta tarea, está permitido el uso de cualquier librería open source o de uso libre, siempre y cuando la licencia de la librería permita los casos de uso propuestos en esta tarea.

Globos

Para el globo de la tarea se recomienda usar globe.gl. Existen alternativas como *CesiumJS*, *WebGL Earth*, *Mapbox GL JS* y *Deck.gl*; pero no han sido testeadas para la realización de la tarea.

Elementos gráficos

Para la generación de la interfaz gráfica, se recomienda (pero no obligatorio) el uso de alguna librería para elementos gráficos. Actualmente, existen varias librerías de este tipo. Algunas de estas se detallan a continuación:

Bootstrap

La librería de elementos gráficos más común. Contiene múltiples elementos gráficos pre- cargados, tales como botones, funciones de *layout*, figuras, banners, entre otros. Posee versiones para múltiples *frameworks*.

- **Ventajas:** múltiples recursos gráficos, así como interacciones y otros elementos de código. Existen versiones para los *frameworks Javascript* más comunes.
- **Desventajas:** Librería con muchos componentes de *Javascript*, lo que puede generar algunas incompatibilidades con algunos *frameworks*. Dependiendo del nivel de personalización que se incluya, puede requerir un proceso de compilación para publicar.



IIC3103 - Taller de Integración

Departamento Ciencia de la Computación
Escuela de Ingeniería
Pontificia Universidad Católica de Chile

Pontificia Universidad Católica

Foundation Framework

Una de las compensaciones de *Bootstrap*. Al igual que *Bootstrap*, contiene múltiples elementos gráficos pre-cargados, así como ciertas interacciones y elementos para *Javascript*.

Pure CSS

Librería compuesta solo por elementos CSS. No contiene ningún componente con *Javascript*.

- **Ventajas:** Extremadamente pequeña en peso, por lo que agrega muy poco *overhead*. No contiene *Javascript* por lo que debería tener compatibilidad con una mayor cantidad de *frameworks* y librerías.
- **Desventajas:** No posee ningún tipo de interacción o animación que requiera código *Javascript*.

Frameworks Javascript

Existen múltiples *frameworks* de *Javascript* que ayudan a construir interfaces responsivas y dinámicas.

Estos *frameworks* posibilitan la generación de aplicaciones complejas, mejoras en performance, *server-side rendering*, arquitecturas de *microfrontends*, manejo de máquinas de estados y una serie de otras funcionalidades, que probablemente no sean utilizadas en el alcance de esta tarea.

Algunos de estos *frameworks* son:

- **React:** *Framework* creado por Facebook, basado en componentes reutilizables. Tiene la versión Native, que permite ser compilada a lenguajes nativos de aplicaciones móviles.
- **Vue:** *Framework* desarrollado en forma independiente con foco en la velocidad y performance de los sitios web.
- **Angular:** *Framework* creado por Google para el desarrollo de aplicaciones web, con foco en PWA, velocidad y performance.

Por lo general, la mayor desventaja de estos *frameworks* son las curvas de aprendizaje, por lo que no se recomienda si no se tiene experiencia con alguno de ellos.



IIC3103 - Taller de Integración

Departamento Ciencia de la Computación
Escuela de Ingeniería
Pontificia Universidad Católica de Chile

Pontificia Universidad Católica

Anexo 3: Detalle de eventos

Emitidos por el *WebSocket*

CATASTROPHIC-FAILURE: Indica que un satélite es destruido. Se desconoce el error.

```
{type: "CATASTROPHIC-FAILURE",  
satellite_id: str}
```

COMM: Permite al operador recibir mensajes vía chat.

```
{type: "COMM",  
message: Message()}
```

DEORBITING: Indica que un satélite cumplió su vida útil y se deja estrellar con la tierra. Es destruido.

```
{type: "DEORBITING",  
satellite_id: str}
```

IN-ORBIT: Indica que un satélite alcanzó su órbita.

```
{type: "IN-ORBIT",  
satellite_id: str,  
altitude: int}
```

LAUNCH: Indica que se lanzó un satélite desde un sitio de lanzamiento.

```
{type: "LAUNCH",  
satellite_id: str,  
launchsite_id: str,  
debris_site: Coordinates()}
```



IIC3103 - Taller de Integración

Departamento Ciencia de la Computación
Escuela de Ingeniería
Pontificia Universidad Católica de Chile

Pontificia Universidad Católica

LAUNCHSITES: Muestra una lista de los sitios de lanzamiento.

```
{type: "LAUNCHSITES",  
launchsites: Launchsite[]}
```

POWER-DOWN: Indica que se disminuyó la potencia del satélite. Disminuye su cobertura.

```
{type: "POWER-DOWN",  
satellite_id: str,  
amount: int}
```

POWER-UP: Indica que se aumentó la potencia del satélite. Aumenta su cobertura.

```
{type: "POWER-UP",  
satellite_id: str,  
amount: int}
```

REQUEST: Solicita una petición al operador.

```
{type: "REQUEST",  
message: Message()}
```

SATELLITES: Muestra una lista de los satélites en órbita y los lanzados y en camino a órbita.

```
{type: "SATELLITES",  
satellites: satellite_id[]}
```

SATELLITE-STATUS: Muestra información del satélite.

```
{type: "SATELLITE-STATUS",  
satellite: Satellite()}
```




IIC3103 - Taller de Integración

Departamento Ciencia de la Computación
Escuela de Ingeniería
Pontificia Universidad Católica de Chile

Pontificia Universidad Católica

Recibidos por el WebSocket

AUTH: Permite al operador acceder al sistema y recibir los comandos del *WebSocket*.

```
{type: "AUTH",  
name: str,  
student_number: str}
```

CHANGE-POWER: Permite cambiar la potencia de un satélite.

```
{type: "CHANGE-POWER",  
ammount: int,  
direction: "UP"|"DOWN"}
```

COMM: Permite al operador mandar mensajes vía chat.

```
{type: "COMM",  
message: str}
```

DESTROY: Permite al operador destruir a distancia un satélite.

```
{type: "DESTROY",  
satellite_id: str}
```

LAUNCHSITES: Permite solicitar una lista de los sitios de lanzamiento.

```
{type: "LAUNCHSITES"}
```

SATELLITES: Permite solicitar una lista de los satélites en órbita y los lanzados y en camino a órbita.

```
{type: "SATELLITES"}
```

SATELLITE-STATUS: Permite solicitar el status de un satélite particular.

```
{type: "SATELLITE-STATUS",  
satellite_id: str}
```



IIC3103 - Taller de Integración

Departamento Ciencia de la Computación
Escuela de Ingeniería
Pontificia Universidad Católica de Chile

Pontificia Universidad Católica

Anexo 4: Detalle de objetos

Objeto tipo Launchsite:

```
{station_id: str,  
name: str,  
country: Country(),  
location: Coordinates(),  
date: Date(),  
content: str}
```

Objeto tipo Message:

```
{station_id: str,  
name: str,  
level: "info"|"warn",  
date: Date(),  
content: str}
```

Objeto tipo Satellite:

```
{satellite_id: str,  
name: str,  
launchsite_origin: station_id,  
launch_date: Date(),  
position: Coordinates(),  
altitude: int,  
mission: str,  
organization: Organization(),  
type: "COM"|"SCI"|"NAV"|"SPY",  
orbital_period: int,  
lifespan: int,  
power: int,  
status: "in-orbit"|"launched"|"lost"}
```

Objeto tipo Coordinates:

```
{lat: float,  
long: float}
```



IIC3103 - Taller de Integración

Departamento Ciencia de la Computación
Escuela de Ingeniería
Pontificia Universidad Católica de Chile

Pontificia Universidad Católica

Objeto tipo Country:

```
{country_code: str[2],  
name: str}
```

Objeto tipo Organization:

```
{name: str,  
country: Country()}
```