IIC3103 - Taller de Integración
Departamento Ciencia de la Computación
Escuela de Ingeniería
Pontificia Universidad Católica

Proyecto – Entrega 3

Introducción

El proyecto del curso es una instancia de trabajo que durará todo el semestre, con entregas acumulativas y evaluaciones formativas.

Esta instancia es de carácter práctico, donde los objetivos principales son la aplicación y consolidación de los conocimientos adquiridos durante el curso. En el proyecto, se simulará una situación realista, donde cada grupo deberá realizar las implementaciones e integraciones necesarias para lograr los objetivos específicos de cada entrega.

Este proyecto busca fomentar el pensamiento crítico, la creatividad y la colaboración, habilidades fundamentales para su futura carrera profesional.

Objetivo del proyecto

El objetivo del proyecto es automatizar un flujo mediante la integración de distintos sistemas que realizan distintas partes de un negocio.

El cumplimiento de los objetivos de cada entrega se medirá mediante indicadores de acuerdo con los resultados esperados.

Contexto

Durante el proyecto, cada grupo simulará ser una cafetería, y deberá ejecutar diferentes procesos que simulen la atención de clientes, fabricación de productos y su respectiva venta.

Durante el transcurso del proyecto, cada grupo deberá implementar integraciones con diferentes sistemas para automatizar los procesos anteriores.

Organización de la cafetería

Productos

Cada cafetería ofrece una serie de productos, que pueden ser obtenidos mediante pedidos a un distribuidor mayorista o preparados en la cocina de la cafetería. Cada tipo de producto tiene un identificador único llamado SKU.

Cada producto tiene las siguientes propiedades:

- SKU: indica a que tipo de producto corresponde
- Fecha de vencimiento: fecha de expiración. Después de esta fecha, el producto no puede ser comercializado
- Ubicación: lugar de la cafetería donde se encuentra el producto
- Refrigeración: indica si el producto debe almacenarse en forma refrigerada o no. Si se almacena un producto que requiere refrigeración en un espacio no refrigerado, su vida útil se reducirá.

Espacios

Cada cafetería cuenta con varios espacios, cada uno con diferentes usos:

- Bodega: sector donde se reciben productos y donde se almacenan los productos que no requieren refrigeración
- Bodega refrigerada: lugar donde se almacenan productos refrigerados y congelados.
- Bodega externa: bodega externa que se puede utilizar en caso de no contar con mayor espacio de almacenamiento. Su uso tiene un costo adicional para cada cafetería.
- Cocina: Lugar donde se preparan alimentos.
- Sector de retiro: espacio donde se pueden retirar productos.

Pedidos

Cada cafetería recibe pedidos por diferentes canales: clientes sentados en mesas, aplicaciones de delivery, sitio propio, etc.

Un pedido consta de una solicitud de varios productos. Cada pedido tiene una fecha y hora máxima de entrega.

Procesos de la cafetería

A continuación, se detallan los principales procesos de cada cafetería. En esta sección se muestran los pasos de cada proceso. Considere que la gestión de estos procesos se encuentra en su mayoría ya implementados en diferentes sistemas. La labor de cada grupo será la de automatizar y gestionar el cumplimiento efectivo de cada uno de los objetivos de negocios.

Proceso de abastecimiento

Mediante este proceso, cada cafetería se abastece de los productos necesarios para la venta a clientes solicitándolos a un distribuidor mayorista. Los productos pueden ser productos listos para la venta, como, por ejemplo, una Coca Cola en lata 350cc, o ingredientes necesarios para la elaboración de otro producto, como, por ejemplo, granos enteros de café tostado, los cuales deben ser molidos para luego producir un café expreso.

El proceso inicia cuando una cafetería (aka un grupo) decide abastecerse de cierto producto. Este proceso puede iniciarse en cualquier momento. Uno de los desafíos del proyecto será determinar la oportunidad y cantidad óptima de abastecimiento para cada producto.

Los productos se solicitan a través del Sistema de gestión de cafetería, realizando una llamada por la API por el servicio correspondiente, indicando el tipo de producto solicitado y la cantidad. Los productos no se reciben en forma automática: al realizar la solicitud, se entregará una fecha y hora estimada de entrega.

A la hora programada de entrega, los productos serán entregados en la bodega de la cafetería. Si no hay espacio disponible, los productos se entregarán en la bodega de almacenamiento externa.

Es importante considerar que la bodega externa tiene un costo adicional por hora de uso. Este costo será de $10 por hora por producto almacenado.

Este costo se factura y paga automáticamente.

Proceso de preparación de alimentos

Este proceso es necesario para preparar alimentos que requieren elaboración en el momento que deben venderse, o que requieran cierto tipo de preparación de la cafetería antes de comercializarlos.

Por ejemplo, los granos enteros de café deben ser molidos antes de preparar un café. Se puede realizar con anticipación o en el momento de preparar un café. También corresponde a la elaboración de un croissant con queso fundido. A diferencia del anterior, este debe prepararse mientras se ordena para garantizar una óptima calidad del producto.

Al igual que el proceso anterior, el momento óptimo de inicio deberá ser determinado por cada grupo.

La elaboración de cada producto tendrá una receta que determina los ingredientes necesarios para su elaboración. Siguiendo los ejemplos anteriores, el café molido requerirá granos de café

entero y el croissant con queso fundido requerirá un croissant, una porción de queso y una porción de mantequilla.

Para iniciar la elaboración, se deben mover todos los productos necesarios desde la bodega y/o bodega refrigerada, según corresponda a cada producto, a la cocina. Una vez que están todos los productos en la cocina, se puede comenzar con la elaboración. La elaboración tiene un tiempo de preparación aleatorio, con un tiempo esperado conocido y diferente para cada elaboración. Todos estos pasos se realizan a través del Sistema de gestión de cafetería.

Una vez terminada la elaboración, el producto terminado queda en la cocina listo para poder ser movido a alguna bodega o al sector de retiro para su entrega.

Proceso de vencimiento de productos

Este proceso se inicia automáticamente en la fecha de vencimiento de un producto por el Sistema de gestión de cafetería. Una vez alcanzada esta fecha, el producto se desecha, se elimina del inventario de la cafetería y no está disponible para cualquier uso: para usarse en la cocina o comercializarse. De la misma forma, deja de ser contabilizado en los espacios utilizados de los distintos sectores de la cafetería.

Proceso de compra a otras cafeterías (b2b)

Cada cafetería se especializará en la producción de ciertos productos específicos, por lo que no podrá fabricar todos los artículos necesarios para su operación. Como resultado, en ocasiones, será necesario que realicen pedidos a otras cafeterías.

Durante este proceso, cada grupo podrá adoptar uno de los siguientes roles:

Comprador

El grupo que necesita adquirir productos de otro. Para ello, debe utilizar el servicio de **Crear Orden de Compra**, proporcionando detalles como el producto requerido, el identificador del grupo vendedor y la fecha límite de la orden.

Luego de haber creado la orden de compra, el grupo comprador deberá notificar al grupo vendedor, usando el servicio **/api/orders** (implementado en la entrega 1).

Vendedor

El grupo que recibe una orden de compra de otro grupo. Esta notificación llegará a través del servicio **/api/orders**, desarrollado en la entrega 1. Al recibir la solicitud, el grupo podrá optar por **aceptar** o **rechazar** la orden de compra.

- **Si acepta la orden**, deberá cumplir con el pedido antes de la fecha límite, despachando los productos mediante el servicio de **Entregar Productos**.

- **Si rechaza la orden**, no se requiere ninguna acción adicional.

Proceso de venta a clientes (b2c)

Los clientes de la cafetería también hacen pedidos en las mesas. Sin embargo, el sistema de pedidos de la cafetería es bastante obsoleto. Después de que el mesero toma el pedido y lo ingresa en el sistema, este se almacena como un archivo en un servidor.

Para poder procesar estos pedidos, será necesario conectarse al servidor utilizando el protocolo sFTP para la transferencia de archivos. De esta manera, podrán recibir los pedidos, prepararlos y entregarlos. Este procedimiento es similar al proceso B2B, con la única diferencia en la forma en que se recibe el pedido.

Las credenciales de acceso al sistema serán enviadas por correo electrónico a cada grupo.

Proceso de facturación

En nuestro sistema de gestión para cafeterías, hemos integrado un flujo de facturación que facilita el proceso de cobro y pago para los clientes. Una vez completado un pedido, el vendedor puede generar una factura para el cliente utilizando la API SOAP de facturación, permitiendo al cliente consultar sus facturas pendientes y realizar pagos de manera rápida y segura.

Además, el sistema permite verificar el saldo bancario de la cafetería mediante el servicio **getBankStatement**, lo cual proporciona un resumen actualizado de las transacciones y el saldo disponible. Esta funcionalidad es útil para llevar un control financiero y optimizar la gestión de pagos.

A continuación, se detallan los pasos para la emisión, consulta y pago de facturas:

Emisión de la factura:

a. Una vez completado el pedido, el vendedor debe emitir una factura para el comprador.
b. Para hacerlo, utiliza el servicio **emitInvoice** de la API SOAP de facturación.

Obtención de facturas pendientes:

c. Después de que la factura ha sido emitida, el comprador puede obtener la lista de sus facturas pendientes.
d. Utiliza el servicio **getInvoices** para obtener los IDs de las facturas emitidas que están en estado "pendiente".

Pago de la factura:

e. El comprador puede pagar una factura específica utilizando el servicio **payInvoice**.
f. Para esto, debe indicar el ID de la factura que desea pagar.

WSDL de la API:

Para acceder a estos servicios, utiliza el siguiente enlace WSDL de la API SOAP de facturación:

https://dev.proyecto.2024-2.tallerdeintegracion.cl/soap/billing?wsdl

Sistemas que intervienen en los procesos

Los sistemas descritos a continuación se encuentran implementados. Cada grupo deberá integrarse a ellos según la forma que cada sistema defina.

Todos los sistemas descritos a continuación operan en dos entornos distintos: uno de producción (o "prod") y uno de pruebas (o "dev"). El entorno de producción es el ambiente real donde las acciones realizadas tienen un impacto directo en el cumplimiento de los objetivos de cada entrega del proyecto. Las métricas utilizadas para evaluar el rendimiento y el logro de los objetivos se medirán exclusivamente en este entorno.

El segundo entorno, de pruebas o desarrollo, se utiliza para realizar experimentos y desarrollar nuevas funcionalidades. A menos que se indique lo contrario, este entorno replica las condiciones del entorno de producción. Los resultados obtenidos y las acciones realizadas en el entorno de desarrollo no se considerarán en las métricas oficiales ni en la evaluación del cumplimiento de los objetivos del proyecto.

Cada entorno está identificado por una URL de integración única, y las credenciales de acceso difieren entre ellos. Se espera que cada equipo implemente su solución considerando la facilidad para cambiar las variables o configuraciones de ambiente según sea necesario.

Sistema de gestión de cafetería

Descripción

Este sistema es el “corazón” de cada cafetería. Este sistema actúa como un ERP, y lleva la gestión de todos los productos que se encuentran en la cafetería:

- Productos almacenados
- Espacio de almacenamiento
- Vencimiento de productos
- Elaboración de productos

Capa de integración

Este sistema expone una API REST. La documentación de esta API se encuentra en:

https://dev.proyecto.2024-2.tallerdeintegracion.cl/coffeeshop/docs/

Sistema de pedidos / órdenes de compra

Descripción

Este sistema maneja todas las órdenes o pedidos recibidos en la cafetería. Todos los pedidos recibidos ingresan a la cafetería por este sistema: los clientes sentados en mesas, pedidos recibidos a través de aplicaciones de delivery, etc.

Cada orden tiene un precio de venta, la cantidad y SKU de los productos ordenados, el canal por donde se recibió y la fecha máxima de entrega.

Capa de integración

El acceso a los pedidos se realiza mediante una API. La descripción de esta API se encuentra en:

https://dev.proyecto.2024-2.tallerdeintegracion.cl/ordenes-compra/docs/

Por otro lado, las notificaciones o recepción de nuevos pedidos se reciben a través de diferentes medios (por ejemplo, b2b o b2c).

Sistema de facturación

Descripción

El sistema de facturación permite generar facturas para todos los pedidos completados satisfactoriamente y recibir facturas por los pedidos de productos realizados.

Adicionalmente, permite gestionar el estado de pago de las facturas y revisar el saldo de caja en base a la facturación emitida y recibida de la cafetería.

Capa de integración

La integración con este sistema se realiza mediante servicios SOAP. El archivo WSDL (Web Services Description Language) de esta API se encuentra disponible en:

https://dev.proyecto.2024-2.tallerdeintegracion.cl/soap/billing?wsdl

Entregas

El proyecto constará de 3 entregas. Las entregas son acumulativas, es decir, se espera que la anterior esté terminada para realizar la siguiente. Quienes no cumplan con los criterios de entregas anteriores, deberán realizar las implementaciones pendientes.

Entrega 3

Durante esta entrega, se agregarán nuevas integraciones. El sistema implementado deberá estar completamente funcional y deberá operar en forma automatizada. Se espera que la operación se mantenga por un tiempo continuo cercano a 72 horas.

Al igual que la entrega anterior, en esta entrega se evaluará el cumplimiento de objetivos de negocio durante el tiempo de entrega.

- Ponderación: 50%
- Fecha de entrega: 21 al 24 de noviembre

Objetivos entrega 3

Durante esta entrega, se espera que se integren al sistema de facturación, y mantengan un saldo positivo en su estado de cuenta.

Integración con sistema de facturación

Cada grupo deberá integrarse con el sistema de facturación y realizar las siguientes acciones:

Como vendedor:

- Canal b2b:
  - Emitir al menos 1000 facturas asociadas a órdenes de compra de otras cafeterías.
- Canal b2b:
  - Emitir al menos 1000 facturas asociadas a órdenes de compra de los clientes dentro de la cafetería.

Como comprador:

- Canal b2b:
  - Pagar al menos 1000 facturas asociadas a órdenes de compra de otras cafeterías.
- Canal b2c:
  - Pagar al menos 1000 facturas asociadas a órdenes de compra de los clientes dentro de la cafetería.

Saldo positivo en su estado de cuenta

Durante esta entrega, se exigirá que cada cafetería mantenga un saldo positivo en su estado de cuenta, es decir, que los ingresos de la cafetería sean mayores a sus gastos. Pueden consultar su saldo usando el servicio **getBankStatement** de la API de facturación.

El sado entregado por este servicio considera el costo de uso por el espacio de almacenamiento externo, a un costo de $10 por hora por producto almacenado.

Sitio web

Además de lo evaluado en la entrega anterior (entrega incremental), deberán incluir:

Sección de facturas

Una sección de su sitio web con información relacionada a las facturas. Esto deberá incluir:

1. Listar las facturas asociadas a su cafetería
    a. Filtrar por fecha (por defecto deben mostrar los 3 últimos días).
    b. Filtrar por facturas emitidas o recibidas
    c. Filtrar por estado de la factura (pendiente / pagada)
    d. En el caso de las facturas pendientes como comprador, un botón para poder pagar la factura. El estado de la factura debería actualizarse inmediatamente o al menos al refrescar la página.
2. Mostrar el saldo actual de la cafetería.
3. Mostrar una serie de tiempo del saldo de la cafetería (saldo / tiempo).

Uptime servidor

Cada grupo deberá tener un uptime del servidor superior al 95% del tiempo de la entrega. Este se medirá mediante la validación del correcto funcionamiento del servicio /api/health implementado en la entrega 1.

Fechas de la entrega

- Habilitación sistema facturación: se encuentra habilitado. El secret es el mismo que la API de cafetería y de órdenes. Se agregó un anexo con ejemplos de cómo conectarse.
- 15/11: Publicación nuevo listado de productos
- 20/11: Habilitación entorno producción

Evaluación

Cada entrega tendrá dos componentes de evaluación:

1. Nota grupal: corresponde a la nota obtenida según la rúbrica y criterios de cada entrega obtenida por el grupo.
2. Nota de evaluación de compañeros: cada estudiante deberá evaluar la contribución realizada por sus compañeros al desarrollo de la entrega.

La nota individual de proyecto corresponderá a la ponderación de ambos componentes. La nota de proyecto corresponderá a la ponderación de la nota individual de cada entrega.

Para la aprobación del curso, se requiere que la nota de evaluación de compañeros y la nota del proyecto sea mayor o igual a 4.0.

IIC3103 - Taller de Integración

Departamento Ciencia de la Computación
Escuela de Ingeniería
Pontificia Universidad Católica

Anexos

IIC3103 - Taller de Integración
Departamento Ciencia de la Computación
Escuela de Ingeniería
Pontificia Universidad Católica

1. Configurando y utilizando el servidor

Cada grupo recibirá un servidor sin nada instalado salvo el sistema operativo y sus componentes básicos. Cada servidor tiene las siguientes características:

- Sistema operativo: Ubuntu 22.04
- Cores: 2 núcleos
- Memoria: 2 GB Ram
- Almacenamiento: 20 GB
- Puertos abiertos: 22 (SSH), 80 (HTTP) y 443 (HTTPS)

Conexión al servidor

La conexión al servidor se debe realizar por la consola mediante SSH con el usuario y contraseña que se proveerá a cada grupo. Para esto, se debe ejecutar en la consola:

```
ssh username@url
```

Donde *username* corresponde al nombre de usuario que se entregará y *url* a la dirección del servidor.

Al ejecutar el comando, se pedirá la contraseña.

Se recomienda fuertemente no cambiar la configuración del SSH para evitar problemas de acceso al servidor. Se prohíbe expresamente cambiar la clave del usuario entregado.

Conexiones entrantes y configuración de Firewall

El firewall de cada servidor está configurado para aceptar conexiones entrantes por los puertos 22 para SSH, 80 para HTTP y 443 para HTTPS. El firewall no permite conexiones por otros puertos distintos a los anteriores.

El firewall está configurado a nivel físico. El firewall de Ubuntu (ufw) no se utiliza por lo que se recomienda fuertemente no cambiar su configuración por defecto para evitar problemas de acceso al servidor. Algunos tutoriales tienen entre sus pasos la configuración del firewall. En estos casos, basta con saltarse esos pasos ya que la configuración de red ya se encuentra lista.

Uso de un servidor web

Se recomienda fuertemente el uso de un sistema de servidor web. Algunas alternativas son Nginx o Apache. El primero es el más utilizado actualmente por su perfomance y simplicidad de uso. También existen otras soluciones más nuevas como Traefik y Caddy. La elección del servidor web quedará a disposición de cada grupo, pero los ejemplos y tutoriales que se mostrarán a continuación se basarán en Nginx.

El servidor web es fundamental para desplegar una solución web de forma robusta y lista para un ambiente productivo. El servidor web se encargará de manejar la comunicación HTTPS y hacer uso del certificado SSL/TLS respectivo para implementar ese canal.

Adicionalmente, el servidor web permite manejar la carga del servidor, gestionar múltiples hilos y coordinar múltiples requests en forma síncrona. Por último, ante errores de código, el servidor web es capaz de restaurar el estado para reponerse ante estos errores.

Ejemplo

A modo de ejemplo, sin servidor web, se podría ejecutar el código de la soluciónccon el comando directo:

```
python app.py
```

o

```
node app.js
```

Pero esto generará una serie de problemas:

- Ante errores, el comando "se cae" y termina, por lo que no se podrán procesar nuevos requests hasta que el comando sea reiniciado. El servidor web maneja los errores en la ejecución y reestablece el servicio en forma automática.
- Estos comandos corren en un solo thread. Esto genera que sólo se pueda procesar un request en paralelo. El servidor web es capaz de mantener varios threads de ejecución en paralelo, además de mantener y coordinar una cola de requests por procesar.
- Por defecto, las ejecuciones en forma directa se comunican por HTTP, por lo que todo el manejo del certificado para HTTPS se deberá realizar por código. La comunicación HTTPS es una funcionalidad común de los servidores web y basta con hacer las configuraciones correctas para comenzar a manejarla

Tutoriales

A continuación se deja una lista de tutoriales de configuración de servidores para diferentes lenguajes y tecnologías de Digital Ocean.

1. How To Install Nginx on Ubuntu 22.04: El siguiente tutorial muestra la instalación y configuración de Nginx en Ubuntu, y con un ejemplo rápido con Python.
2. How To Serve Flask Applications with Gunicorn and Nginx on Ubuntu 22.04: Este tutorial muestra en detalle la instalación de Python con Flask y Nginx. Incluye la generación de certificados SSL/TLS para comunicación con HTTPS.

3. How To Set Up a Node.js Application for Production on Ubuntu 22.04: Muestra la instalación en detalle de Node y Nginx. No incluye la configuración de HTTPS (ver anexo siguiente).
4. How To Install PostgreSQL on Ubuntu 22.04 [Quickstart]: instalación rápida de PostgreSQL en servidor Ubuntu.
5. Otros tutoriales para Ubuntu 22.04: Listado de todos los tutoriales para Ubuntu 22.04 en Digital Ocean.

Importante

En varios tutoriales se menciona la configuración del Firewall de Ubuntu (ufw). Como se comenta en la sección "Conexiones entrantes y configuración de Firewall", no se deben realizar cambios a la configuración ya que la configuración del Firewall se realiza en forma externa al servidor. Cambios en estas configuraciones pueden generar bloqueos de acceso al servidor.

2. Configurar HTTPS

Todos los servicios deberán estar expuestos por protocolo HTTPS. Para crear un certificado HTTPS, se recomienda generar un certificado con Let's Encrypt, herramienta gratuita para generación de certificados SSL/TLS.

Adicionalmente, existe Certbot, una herramienta para la CLI que simplifica y automatiza algunos de los pasos necesarios para la creación del certificado.

Para poder utilizar el certificado, es necesario configurar el uso de un servidor web como Apache o Nginx. Se recomienda utilizar el segundo, ya que es más sencillo que el primero.

Para configurar el certificado con Certbot, necesitarás:

- Usuario con sudo y acceso SSH: los usuarios de cada grupo para el acceso al servidor tienen sudo. El acceso al servidor se realiza por SSH.
- Puertos 80 y 443 abiertos: estos puertos ya se encuentran abiertos y habilitados en el Firewall de la universidad.
- Una URL: cada servidor tiene su propia URL.

Sobre la renovación de certificados

Todos los certificados de Let's Encrypt tienen una duración de 90 días. Si se configura un certificado los primeros días de abril, este será válido hasta los primeros días de julio, después de la entrega final, por lo que no será necesario configurar la renovación automática de estos.

Tutoriales para la configuración

Tutorial de Certbot

Link: https://certbot.eff.org/instructions?ws=nginx&os=ubuntufocal

Este tutorial describe el paso a paso para la habilitación del certificado con Ubuntu y Nginx. También están los tutoriales para otras tecnologías de servidores web.

Tutorial de Digital Ocean

Link: https://www.digitalocean.com/community/tutorials/how-to-secure-nginx-with-let-s-encrypt-on-ubuntu-22-04

Este tutorial es similar al anterior, pero se deben tener algunas consideraciones, cómo por ejemplo, no cambiar la configuración del Firewall de Ubuntu (ufw) ya que la configuración de Firewall se administra a nivel de infraestructura y no de sistema operativo.

3. Interactuando con una API SOAP

Para conectarse a una API SOAP, se requiere un cliente SOAP. Para Python, el más popular es Zeep, y para Node, la librería SOAP.

La librería utiliza autenticación básica (*basic auth*) con usuario y contraseña, donde el usuario es el número de grupo y la contraseña es el mismo secreto utilizado para generar tokens en los servicios de Bodega y Órdenes de compra.

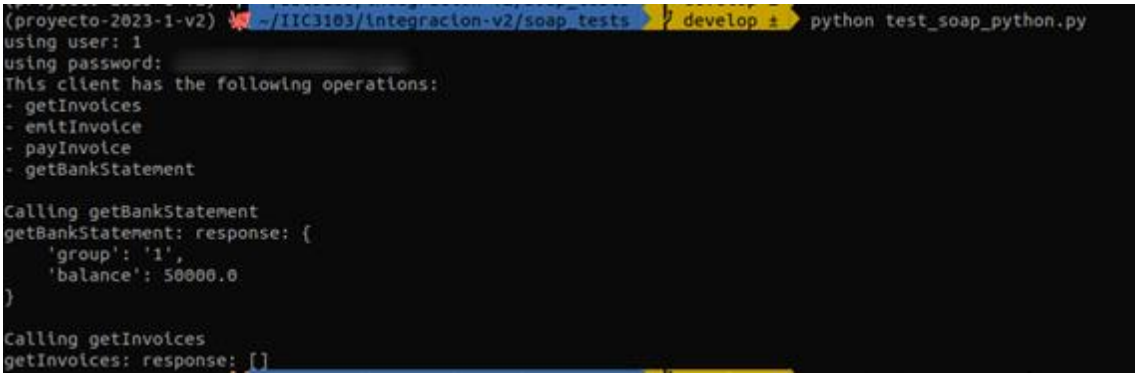
A continuación, se muestra ejemplos para ambas librerías. En ambos ejemplos se debe configurar el usuario y contraseña como variable de ambiente.

Python

```python
from zeep import Client
from zeep.wsse.username import UsernameToken
import os

url = 'https://dev.proyecto.2024-1.tallerdeintegracion.cl/soap/billing?wsdl'

user = os.getenv('SOAP_USER', '')
password = os.getenv('SOAP_PASSWORD', '')

print("using user:", user)
print("using password:", password)

wsse_auth = UsernameToken(user, password)

client = Client(url, wsse=wsse_auth)

print('This client has the following operations:')
for op in client.service._operations.keys():
    print(f' - {op}')

print()

try:
    print("Calling getBankStatement")
    response = client.service.getBankStatement()
    print('getBankStatement: response:', response)

except Exception as e:
    print(f'Error while getting bank statement: {e.message} - {e.detail.text}')

print()

try:
    print("Calling getInvoices")
    response = client.service.getInvoices(status='pending', side='client', fromDate='2024-06-10', toDate='2026-06-11')
    print('getInvoices: response:', response)

except Exception as e:
    print(f'Error while emiting invoice: {e.message} - {e.detail.text}')
```

Resultado esperado

Resultado esperado al ejecutar el ejemplo:



Node

```
var soap = require('soap');

var url = 'https://dev.proyecto.2024-1.tallerdeintegracion.cl/soap/billing?wsdl';
let user = process.env.SOAP_USER;
let password = process.env.SOAP_PASSWORD;
console.log("using user: " + user);
console.log("using password: " + password);
var wsSecurity = new soap.WSSecurity(user, password, {});

function getBankStatementAsync() {
  return new Promise((resolve, reject) => {
    soap.createClient(url, {}, function(err, client) {
      client.setSecurity(wsSecurity);
      console.log("This client has the following operations:");
      console.log(client.describe());
      console.log("\n");
      console.log("Calling getBankStatement operation:");
      client.getBankStatement({}, function(err, result) {
        resolve(result); // resolve promise
      });
    });
  });
};

function getInvoicesAsync(status, side, fromDate, toDate) {
  return new Promise((resolve, reject) => {
    soap.createClient(url, {}, function(err, client) {
      client.setSecurity(wsSecurity);
      var getInvoicesArgs = {
        status,
        side,
        fromDate,
        toDate
      };

      console.log("Calling getInvoices operation:");
      client.getInvoices(getInvoicesArgs, function(err, result) {
        if (result) {
          resolve(result.BillingDetails); // resolve promise
        }
```

```
      resolve([]); // resolve promise
    });
  });
 });
};
```

```
// Now you can use await to get the result
async function main() {
 console.log("\n");
 try {
    let result = await getBankStatementAsync();
    console.log(result);
 } catch(error) {
    console.error("ERROR:" + error);
 }

 console.log("\n");

 try {
    let result = await getInvoicesAsync('pending', 'client', '2024-06-10', '2024-06-11');
    console.log(result);
 } catch(error) {
    console.error("ERROR:" + error);
 }
};

main();
```

Resultado esperado

Resultado esperado al ejecutar el ejemplo:

```
(proyecto-2023-1-v2) ~/IIC3103/integracion-v2/soap_tests develop ± node test_soap_node.js
using user: 1
using password:

This client has the following operations:
{
  BillingService: {
    BillingServiceSoapPort: {
      getInvoices: [Object],
      emitInvoice: [Object],
      payInvoice: [Object],
      getBankStatement: [Object]
    }
  }
}

Calling getBankStatement operation:
{ BankStatement: { group: '1', balance: 50000 } }

Calling getInvoices operation:
[]
```

Versiones del documento

- **Versión 1:** 4/11/2024
- Versión 2: 14/11/2024

- Se agrega anexo con uso de API SOAP