

Instalación

Para usar el SDK, primero debes generar el paquete NuGet. Para ello, sigue los siguientes pasos:

Generar un paquete NuGet

1. Clona el repositorio en tu máquina local.
2. Abre una terminal y navega hasta la carpeta del proyecto donde se encuentra el archivo de solución (`.sln`).
3. Ejecuta el siguiente comando para generar el paquete NuGet en configuración Release:

```
dotnet pack --configuration Release
```

Esto creará el archivo `.nupkg` en el directorio `bin/Release`.

Una vez hayas generado el paquete NuGet (`.nupkg`), puedes añadirlo a tu proyecto de la siguiente manera:

Usando Visual Studio

1. Copia el archivo `.nupkg` generado al directorio de tu elección en tu PC.
2. Abre Visual Studio y navega a **Herramientas > Administrador de paquetes NuGet > Configuración del Administrador de paquetes**.
3. En la sección **Orígenes de paquetes**, haz clic en el botón **+** para agregar un nuevo origen.
4. Asigna un nombre al origen (por ejemplo, `LocalPackages`) y selecciona la ruta del directorio donde copiaste el archivo `.nupkg`.
5. Haz clic en **Aceptar** para guardar los cambios.
6. Abre tu proyecto en Visual Studio.
7. Haz clic derecho sobre el proyecto y selecciona **Administrar paquetes NuGet**.
8. Selecciona el origen que acabas de agregar (`LocalPackages`) en el menú desplegable **Origen de paquete**.
9. Busca el paquete NuGet que generaste e instálalo.

Usando la línea de comandos

1. Copia el archivo `.nupkg` generado al directorio de tu elección en tu PC.
2. Abre una terminal y navega hasta la carpeta de tu proyecto.
3. Configura el origen del paquete NuGet ejecutando el siguiente comando:

```
dotnet nuget add source <ruta al directorio con el .nupkg> -n LocalPackages
```

4. Añade el paquete NuGet a tu proyecto ejecutando el siguiente comando:

```
dotnet add package <nombre del paquete>
```

Esto configurará el origen del paquete y añadirá el paquete NuGet a tu proyecto.

Servicio de Autenticación

El servicio de autenticación permite obtener un token de acceso para interactuar con los servicios de LUNDERO. Para ello, debes proporcionar tus credenciales de acceso.

Detalles de integración

ClienteAutenticacion

LoginAsync(LoginSolicitud login)

Este método realiza una solicitud de inicio de sesión asincrónica para obtener un token de autenticación.

- **Parámetros:**
 - `login` ([LoginSolicitud](#)): Contiene el usuario y la clave.
- **Retorno:**
 - `Task<[LoginRespuesta]>` ([LoginRespuesta](#)): Una tarea que representa la operación asincrónica. El resultado contiene el token de autenticación. Si LoginRespuesta es null es por que el usuario o contraseña son inválidos.

LoginSolicitud

- **Usuario** (string): Nombre de usuario.
- **Clave** (string): Contraseña.

LoginRespuesta

- **Token** (string): Token para utilizarlo en los diferentes servicios.

Ejemplo

```
using LunderoSDK.Autenticacion.NetStandard;
using LunderoSDK.Autenticacion.NetStandard.Solicitudes;

var configuracion = new ClienteAutenticacionConfig
{
    Url = "https://apisigerproducecloud.iamauta.com",
};

var login = new LoginSolicitud
{
    Usuario = "usuarioABC",
    Clave = "claveABC"
};
```

```
var cliente = new ClienteAutenticacion(configuracion);  
  
var respuesta = await cliente.LoginAsync(login);  
  
var authToken = respuesta.Token;
```

Servicio de Encuestas

Permite interactuar con los servicios de encuestas de Lundero, para poder utilizarlos se necesita el token obtenido con el cliente de autenticación (Login):

- [Obtener la lista de encuestas por ruc.](#)
- [Obtener el detalle de una encuesta.](#)
- [Guardar una encuesta.](#)

Obtener la lista de encuestas por ruc

Este servicio obtiene las encuestas asociadas a un RUC.

Detalles de integración

ClienteEncuesta

`ObtenerEncuestasPorRucAsync(string ruc)`

Este método obtiene una lista de encuestas asociadas a un RUC de manera asincrónica.

- **Parámetros:**
 - `ruc` (string): El RUC para el cual se desean obtener las encuestas.
- **Retorno:**
 - `Task<List<EncuestaRespuesta>>` ([EncuestaRespuesta](#)): Una tarea que representa la operación asincrónica. El resultado contiene una lista de encuestas.

EncuestaRespuesta

- **IdFormulario** (int): Identificador del formulario de la encuesta.
- **Version** (int): Versión del formulario de la encuesta.
- **IdEjecucion** (int): Identificador de la ejecución de la encuesta.
- **CodigoAlternativo** (string): Código alternativo de la encuesta.
- **Nombre** (string): Nombre de la encuesta.
- **Descripcion** (string): Descripción de la encuesta.
- **Cuerpo** (string): Cuerpo de la encuesta.
- **TipoFormulario** (string): Tipo de formulario de la encuesta.
- **FormularioAnterior** (string): Formulario anterior de la encuesta.
- **VersionAnterior** (int?): Versión anterior del formulario de la encuesta.
- **FechaRegistro** (string): Fecha de registro de la encuesta.
- **EstadoFormulario** (string): Estado del formulario de la encuesta.
- **FlagConsultaDinamica** (bool?): Valor que indica si la consulta es dinámica.
- **NumPreguntas** (int): Número de preguntas de la encuesta.

- **DuracionEncuesta** (int): Duración de la encuesta.
- **IdTarea** (int): Identificador de la tarea asociada a la encuesta.
- **IdTrabajo** (int): Identificador del trabajo asociado a la encuesta.
- **IdFlujo** (int): Identificador del flujo asociado a la encuesta.
- **IdFormularioMaster** (int): Identificador del formulario maestro de la encuesta.
- **IdParticipante** (int): Identificador del participante de la encuesta.
- **IdPrioridadTarea** (int): Identificador de la prioridad de la tarea asociada a la encuesta.
- **IdEstructura** (int): Identificador de la estructura asociada a la encuesta.
- **IdEstructuraSec** (int): Identificador de la estructura secundaria asociada a la encuesta.
- **Contenido** (string): Contenido de la encuesta.
- **NumeroTrabajo** (string): Número de trabajo asociado a la encuesta.
- **EncuestaCompletada** (int): Valor que indica si la encuesta está completada.

Ejemplo

```
using LunderoSDK.Encuesta.NetStandard;
using LunderoSDK.Encuesta.NetStandard.Solicitudes;

var ruc = "20605565655";

var configuracion = new ClienteEncuestaConfig
{
    Url = "https://apisigerproducecloud.iamauta.com",
    Autorizacion = "token"
};

var cliente = new ClienteEncuesta(configuracion);

var respuesta = await cliente.ObtenerEncuestasPorRucAsync(ruc);
```

Obtener el detalle de una encuesta

Este servicio obtiene las preguntas y alternativas de una determinada encuesta, para esto se necesita enviar los siguientes datos:

- El identificador del formulario (IdFormulario).
- La versión del formulario (Version).
- El identificador de la ejecución (IdEjecucion).

Detalles de integración

ClienteEncuesta

`ObtenerEncuestaAsync(EncuestaSolicitud encuesta)`

Este método obtiene los detalles de una encuesta específica de manera asincrónica.

- **Parámetros:**
 - **encuesta** ([EncuestaSolicitud](#)): La solicitud que contiene los parámetros necesarios para obtener la encuesta.
- **Retorno:**
 - **Task<[EncuestaDetalleRespuesta]>** ([EncuestaDetalleRespuesta](#)): Una tarea que representa la operación asincrónica. El resultado contiene los detalles de la encuesta.

EncuestaSolicitud

- **IdFormulario** (int): Identificador del formulario de la encuesta.
- **Version** (int): Versión del formulario de la encuesta.
- **IdEjecucion** (int): Identificador de la ejecución de la encuesta.

EncuestaDetalleRespuesta

- **Encuesta** ([Encuesta](#)): Detalles de la encuesta.

Encuesta

- **IdFormulario** (int): Identificador del formulario de la encuesta.
- **Version** (int): Versión del formulario de la encuesta.
- **CodigoAlternativo** (string): Código alternativo de la encuesta.
- **Nombre** (string): Nombre de la encuesta.
- **Descripcion** (string): Descripción de la encuesta.
- **FechaRegistro** (string): Fecha de registro de la encuesta.
- **NumPreguntas** (int): Número de preguntas de la encuesta.
- **DuracionEncuesta** (int): Duración de la encuesta.
- **DescripcionEncuesta** (List): Descripción de la encuesta.
- **PreguntasEncuesta** (List<[PreguntasEncuesta](#)>): Lista de preguntas de la encuesta.

PreguntasEncuesta

- **Nombre** ([ContenedorValue](#)): Nombre de la pregunta.
- **Tipo** ([ContenedorValue](#)): Tipo de la pregunta.
- **Items** ([Alternativa](#)): Contenedor de las opciones de la pregunta.

ContenedorValue

- **Value** (string): valor.

Alternativa

- **Value** (string): Valor de la alternativa.

- **Options** (List<[Opcion](#)>): Lista de opciones de la pregunta.

Opcion

- **Id** (string): Identificador de la opción.
- **Label** (string): Etiqueta de la opción.

Ejemplo

```
using LunderoSDK.Encuesta.NetStandard;
using LunderoSDK.Encuesta.NetStandard.Solicitudes;

var configuracion = new ClienteEncuestaConfig
{
    Url = "https://apisigerproducecloud.iamauta.com",
    Autorizacion = "token"
};

var encuestaSolicitud = new EncuestaSolicitud
{
    IdFormulario = 940,
    Version = 1,
    IdEjecucion = 5052
};

var cliente = new ClienteEncuesta(configuracion);

var respuesta = await cliente.ObtenerEncuestaAsync(encuestaSolicitud);
```

Guardar una encuesta

Este servicio permite guardar las preguntas y alternativas de la encuesta además de los datos enviados en los servicios anteriores.

Detalles de integración

ClienteEncuesta

GuardarEncuestaAsync(EncuestaAGuardarSolicitud encuesta)

Este método guarda una encuesta de manera asincrónica.

- **Parámetros:**
 - **encuesta** ([EncuestaAGuardarSolicitud](#)): La solicitud que contiene los datos de la encuesta a guardar.

- **Retorno:**
 - `Task<List<EncuestaGuardadaRespuesta>>` ([EncuestaGuardadaRespuesta](#)): Una tarea que representa la operación asincrónica. El resultado contiene una lista de respuestas de encuestas guardadas.

EncuestaAGuardarSolicitud

- **ConCopia** (List): Lista de destinatarios con copia.
- **DataFormulario** (List<[DataFormulario](#)>): Lista de datos del formulario.
- **RespuestaEncuesta** (List<[RespuestaEncuesta](#)>): Lista de respuestas de la encuesta.
- **TrabajoEjecucion** ([TrabajoEjecucion](#)): Detalles de la ejecución del trabajo.
- **EjecucionTarea** ([EjecucionTarea](#)): Detalles de la ejecución de la tarea.
- **TrabajoArchivos** (List): Lista de archivos del trabajo.
- **IdTareaAnterior** (int): Identificador de la tarea anterior.
- **IdEstructura** (int): Identificador de la estructura.

DataFormulario

- **Contenido** ([ContenidoDataFormulario](#)): Contenido del formulario.
- **IdFormularioMaster** (int): Identificador del formulario maestro.
- **ContenidoStr** (string): Obtiene una representación en cadena del contenido del formulario.

ContenidoDataFormulario

- **Campos** (Dictionary<string, object>): Campos del formulario.
- **Adjuntos** (List): Lista de adjuntos del formulario.
- **EstadoFormulario** (Dictionary<string, object>): Estado del formulario.

RespuestaEncuesta

- **Num** (string): Número de la respuesta.
- **Respuesta** (List): Lista de respuestas.

TrabajoEjecucion

- **IdEjecucion** (int): Identificador de la ejecución.
- **IdTrabajo** (int): Identificador del trabajo.

EjecucionTarea

- **Comentario** (string): Comentario de la tarea.
- **Flujos** (List): Lista de flujos de la tarea.
- **IdEjecucion** (int): Identificador de la ejecución.
- **TipoDerivacion** (string): Tipo de derivación de la tarea.
- **IdTarea** (int?): Identificador de la tarea.
- **IdParticipante** (int): Identificador del participante.

- **PrioridadTarea** (int): Prioridad de la tarea.
- **Estructuras** (List): Lista de estructuras de la tarea.

EncuestaGuardadaRespuesta

- **ReferenciaVersion** (string): Referencia de la versión.
- **IdEjecucion** (int): Identificador de la ejecución.
- **ReferenciaCodAlterno** (string): Referencia del código alternativo.
- **IdTarea** (string): Identificador de la tarea.
- **IdArchivo** (int): Identificador del archivo.
- **IdReferencia** (string): Identificador de la referencia.
- **Nombre** (string): Nombre de la encuesta.

Ejemplo

```
using LunderoSDK.Encuesta.NetStandard;
using LunderoSDK.Encuesta.NetStandard.Solicitudes;

var configuracion = new ClienteEncuestaConfig
{
    Url = "https://apisigerproducecloud.iamauta.com",
    Autorizacion = "token"
};

var contenido = new ContenidoDataFormulario
{
    Campos = new Dictionary<string, object>
    {
        { "etiqueta", "" },
        {
            "en_su_empresa_se_valora_y_se_realizan_esfuerzos_para_contar_con_mas_y_mejor_calidad_de_datos_yo_informacion_.0", true }
    }
};

var encuestaAGuardar = new EncuestaAGuardarSolicitud
{
    DataFormulario = new List<DataFormulario>
    {
        new DataFormulario
        {
            Contenido = contenido,
            IdFormularioMaster = 582
        }
    },
};
```

```

RespuestaEncuesta = new List<RespuestaEncuesta>
{
    new RespuestaEncuesta
    {
        Num = "1",
        Respuesta = new List<string>
        {
            "Hay una ALTA disposición al cambio para implementar tecnologías digitales,
            todos están comprometidos y entusiasmados con la adopción de nuevas tecnologías"
        }
    }
},
TrabajoEjecucion = new TrabajoEjecucion
{
    IdEjecucion = 5052,
    IdTrabajo = 998
},
EjecucionTarea = new EjecucionTarea
{
    Comentario = "Comentario",
    Flujos = new List<int> { 1819 },
    IdEjecucion = 5052,
    TipoDerivacion = "D",
    IdTarea = null,
    IdParticipante = 7756,
    PrioridadTarea = 32047,
    Estructuras = new List<int> { 2437 }
},
IdTareaAnterior = 5886,
IdEstructura = 2436
};

var cliente = new ClienteEncuesta(configuracion);

var respuesta = await cliente.GuardarEncuestaAsync(encuestaAGuardar);

```