



PRÓXIMOS PASOS - DEPLOYMENT ENDPOINT

Estado actual: ☒ Código completo y listo para deployment **Pendiente:** Elegir plataforma + Deployment + Configuración de credenciales + Testing

⚠ IMPORTANTE: Limitaciones de Free Tier

El endpoint usa **Playwright** que tarda ~10-15 segundos en ejecutar el login completo.

Vercel Free Tier

- **✗ Timeout: 10 segundos** → Playwright hace timeout
- **✗ Memory: 1024 MB** → Justo para Playwright
- ☒ Deploy automático desde GitHub
- ☒ Setup muy simple

Railway.app (RECOMENDADA)

- ☒ **Sin límites de timeout** → Playwright funciona sin problemas
 - ☒ **\$5 USD gratis/mes** (crédito, no recursos limitados)
 - ☒ Deploy automático desde GitHub
 - ☒ Setup similar a Vercel
 - **Estimado:** \$5/mes alcanza para uso moderado
-



ELIGE TU PLATAFORMA

Opción A: Railway.app (Recomendada para Free Tier)

Opción B: Vercel (Requiere Hobby Plan \$20/mes)



CHECKLIST DE DEPLOYMENT

☒ Fase 1: Commit y Deploy (COMPLETADA por Claude)

- ☒ **1.1. Código creado**
 - `api/crear-factura.js`, `api/test-login.js`
 - `api/utils/browserLogin.js`, `api/utils/buildXMLPayload.js`
 - `vercel.json`, `package.json` actualizado
 - Documentación completa
- ☒ **1.2. Commit y push**
 - Commit: `505cb17`
 - Message: "feat: Endpoint Vercel para facturación con Playwright..."
 - Pusheado a `origin/main`

- ☐ **1.3. Verificar auto-deploy en Vercel** (VOS tenés que hacer esto)
 - Ir a <https://vercel.com/dashboard>
 - Ver que el proyecto se está deployando (ícono amarillo girando)
 - Esperar a que termine (ícono verde ☒)
 - **Copiar URL del deployment:** <https://TU-PROYECTO.vercel.app>

Si NO está conectado a Vercel:

- Click en "Add New" → "Project"
- Importar repo de GitHub/GitLab
- Framework: Vite
- Root Directory: [./test-imprimir-pdf](#)
- Deploy

☒ Fase 2: Configurar Credenciales (3 minutos)

⚠ **CRÍTICO:** Sin estas variables el endpoint NO funciona.

- ☐ **2.1. Ir a configuración de variables**
 - Vercel Dashboard → Tu proyecto → **Settings** → **Environment Variables**
- ☐ **2.2. Agregar primera variable**
 - Name: [XUBIO_USERNAME](#)
 - Value: [martin.lombardi@gmail.com](#)
 - Environments: ☒ Production | ☒ Preview | ☒ Development
 - Click "**Save**"
- ☐ **2.3. Agregar segunda variable**
 - Name: [XUBIO_PASSWORD](#)
 - Value: [Corvus"22](#)
 - Environments: ☒ Production | ☒ Preview | ☒ Development
 - Click "**Save**"
- ☐ **2.4. REDEPLOY (obligatorio)**
 - Ir a tab "**Deployments**"
 - Click en el deployment más reciente (arriba de todo)
 - Click en menú "..." (tres puntos)
 - Click en "**Redeploy**"
 - Esperar que termine (~2 minutos)

☒ Fase 3: Testing (5 minutos)

Reemplazar [TU-PROYECTO.vercel.app](#) con tu URL real.

Test 1: Login

```
curl -X POST https://TU-PROYECTO.vercel.app/api/test-login
```

 **Tiempo esperado:** 8-12 segundos

☒ **Response exitoso:**

```
{
  "success": true,
  "message": "Login exitoso",
  "data": {
    "cookiesCount": 5,
    "cookiesValid": true,
    "cookieHeader": "...",
    "cookies": [...]
  }
}
```

 **Errores comunes:**

Error	Causa	Solución
"Missing credentials"	Variables no configuradas	Revisar Fase 2, hacer redeploy
"Login falló"	Credenciales incorrectas	Verificar password (tiene comillas)
Function timeout	Playwright tardó mucho	Esperar y reintentar

Test 2: Crear Factura

Opción A - Con curl:

```
curl -X POST https://TU-PROYECTO.vercel.app/api/crear-factura \
-H "Content-Type: application/json" \
-d '{
  "clienteId": 8157173,
  "clienteNombre": "2MCAMPO",
  "provinciaId": 1,
  "provinciaNombre": "Buenos Aires",
  "localidadId": 147,
  "localidadNombre": "Saladillo",
  "cantidad": 1
}'
```

Opción B - Con Postman:

- 1. Nuevo request POST

2. URL: `https://TU-PROYECTO.vercel.app/api/crear-factura`
3. Headers: `Content-Type: application/json`
4. Body (raw JSON):

```
{
  "clienteId": 8157173,
  "clienteNombre": "2MCAMPO",
  "provinciaId": 1,
  "provinciaNombre": "Buenos Aires",
  "localidadId": 147,
  "localidadNombre": "Saladillo",
  "cantidad": 1
}
```

5. Send

🕒 **Tiempo esperado:** 10-15 segundos

☑ **Response exitoso:**

```
{
  "success": true,
  "message": "Factura creada exitosamente",
  "data": {
    "transaccionId": "67750488",
    "numeroDocumento": "A-00004-00001680",
    "total": 593.9,
    "pdfUrl": "https://xubio.com/NXV/transaccion/ver/67750488",
    "cotizacion": 1455,
    "cantidad": 1
  }
}
```

☑ **Validar en Xubio:**

- Ir a `https://xubio.com`
- Ver que la factura se creó con el TransaccionID retornado
- Abrir PDF desde `pdfUrl` de la response



OPCIÓN A: DEPLOYMENT EN RAILWAY.APP (RECOMENDADA)

☑ **Ventajas sobre Vercel Free Tier**

- Sin límites de timeout (Playwright funciona perfectamente)
- \$5 USD gratis/mes (suficiente para uso moderado)
- Deploy automático desde GitHub
- Variables de entorno igual que Vercel

Fase 1: Setup Inicial (5 minutos)

1.1. Crear cuenta en Railway

- ☐ Ir a <https://railway.app>
- ☐ Click en **"Start a New Project"**
- ☐ Login con GitHub

1.2. Crear proyecto desde GitHub

- ☐ Click en **"Deploy from GitHub repo"**
- ☐ Seleccionar tu repositorio
- ☐ Railway detecta automáticamente Node.js
- ☐ Click en **"Deploy Now"**

1.3. Configurar root directory (si es necesario)

Si el proyecto está en subdirectorio:

- ☐ Click en el servicio deployado
- ☐ Tab **"Settings"**
- ☐ Sección **"Build"**
- ☐ **Root Directory:** Dejar vacío (usar root del repo)
- ☐ **Start Command:** `node api/crear-factura.js` (Railway detecta automáticamente)

Fase 2: Configurar Variables de Entorno (3 minutos)

- ☐ Click en tu servicio deployado
- ☐ Tab **"Variables"**
- ☐ Click en **" + New Variable"**

Agregar primera variable:

- ☐ Variable: `XUBIO_USERNAME`
- ☐ Value: `martin.lombardi@gmail.com`
- ☐ Click **"Add"**

Agregar segunda variable:

- ☐ Variable: `XUBIO_PASSWORD`
- ☐ Value: `Corvus"22`
- ☐ Click **"Add"**

Railway redeploya automáticamente después de agregar variables.

Fase 3: Obtener URL del Endpoint

- ☐ Click en tu servicio

- ☐ Tab **"Settings"**
- ☐ Sección **"Networking"**
- ☐ Click en **"Generate Domain"**
- ☐ Copiar URL generada (ejemplo: `tu-proyecto-production.up.railway.app`)

Tu endpoint estará disponible en:

```
https://tu-proyecto-production.up.railway.app/api/test-login  
https://tu-proyecto-production.up.railway.app/api/crear-factura
```

Fase 4: Testing (5 minutos)

Test 1: Login

```
curl -X POST https://tu-proyecto-production.up.railway.app/api/test-login
```

☒ Response esperado:

```
{  
  "success": true,  
  "message": "Login exitoso",  
  "data": {  
    "cookiesCount": 5,  
    "cookiesValid": true,  
    "cookieHeader": "...",  
    "cookies": [...]  
  }  
}
```

Test 2: Crear Factura

```
curl -X POST https://tu-proyecto-production.up.railway.app/api/crear-factura \  
-H "Content-Type: application/json" \  
-d '{  
  "clienteId": 8157173,  
  "clienteNombre": "2MCAMPO",  
  "provinciaId": 1,  
  "provinciaNombre": "Buenos Aires",  
  "localidadId": 147,  
  "localidadNombre": "Saladillo",  
  "cantidad": 1  
}'
```

Monitoreo de Uso (importante para free tier)

- ☐ Tab "**Metrics**" → Ver CPU/RAM usage
- ☐ Tab "**Deployments**" → Ver logs en tiempo real
- ☐ Tab "**Usage**" → Ver crédito restante del mes

Estimado de consumo:

- Login + Factura: ~\$0.01 por request
 - \$5/mes ≈ 500 requests/mes
 - Si usás más, Railway te avisa antes de cobrar
-

Ver Logs

Opción A - Dashboard:

1. Click en tu servicio
2. Tab "**Deployments**"
3. Click en deployment activo
4. Ver logs en tiempo real

Opción B - CLI:

```
npm i -g @railway/cli
railway login
railway logs
```

🔑 OPCIÓN B: DEPLOYMENT EN VERCEL (Plan Pago)

⚠ **IMPORTANTE:** Vercel Free Tier **NO funciona** con Playwright (timeout de 10s). Necesitás **Hobby Plan (\$20/mes)** para tener 60s de timeout.

Si tenés Hobby Plan, seguir estas instrucciones:

Fase 1: Actualizar vercel.json

El repo ya tiene **vercel.json** configurado para free tier (10s timeout). Si tenés Hobby Plan, actualizalo:

```
{
  "functions": {
    "api/**/*.js": {
      "memory": 3008,
      "maxDuration": 60
    }
  }
}
```

Fase 2: Deploy en Vercel

- ☐ Ir a <https://vercel.com/dashboard>
 - ☐ Click en **"Add New"** → **"Project"**
 - ☐ Importar repo de GitHub
 - ☐ Framework: **Other**
 - ☐ Root Directory: **/** (root del repo)
 - ☐ Click **"Deploy"**
-

Fase 3: Configurar Variables de Entorno

- ☐ Vercel Dashboard → Tu proyecto → **Settings** → **Environment Variables**

Agregar variables:

- ☐ Name: **XUBIO_USERNAME**, Value: **martin.lombardi@gmail.com**
- ☐ Environments: ☒ Production | ☒ Preview | ☒ Development
- ☐ Name: **XUBIO_PASSWORD**, Value: **Corvus"22**
- ☐ Environments: ☒ Production | ☒ Preview | ☒ Development

Redeploy obligatorio:

- ☐ Tab **"Deployments"**
 - ☐ Click en deployment más reciente
 - ☐ Menú **"..."** → **"Redeploy"**
-

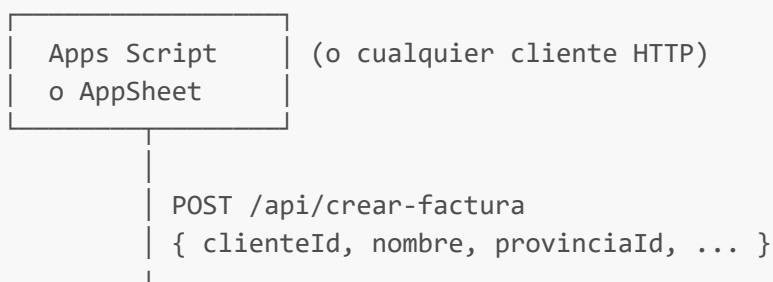
Fase 4: Testing

Igual que Railway, pero usando tu URL de Vercel:











```
curl -X POST https://tu-proyecto.vercel.app/api/test-login
```

FLUJO COMPLETO

Diagrama Visual




Vercel Serverless Function (Node.js + Playwright)

1.  Login Automático
 - Playwright abre Chrome headless
 - Va a xubio.com
 - Redirige a Visma Connect
 - Completa formulario:
 - Username: martin.lombardi@...
 - Password: Corvus"22
 - Click "Iniciar sesión"
 - Espera redirect a xubio.com
 - Extrae cookies de sesión
 -  ~8 segundos
2.  Obtener Cotización USD
 - API pública BCRA
 -  ~1 segundo
3.  Construir XML
 - Template GOLD
 - Datos dinámicos del cliente
 -  ~0.5 segundos
4.  POST a Xubio
 - URL: /NXV/DF_submit
 - Body: XML URL-encoded
 - Headers: Cookies de sesión
 -  ~3 segundos
5.  Parsear Response
 - Extraer TransaccionID
 - Extraer NumeroDocumento
 - Construir PDF URL
 -  ~0.5 segundos

Response JSON

```
{ success,  
  transaccionId  
  pdfUrl, ... }
```

 LATENCIA TOTAL: ~10-15 segundos

INPUT (lo que enviás):

```
{
  "clienteId": 8157173,
  "clienteNombre": "2MCAMPO",
  "provinciaId": 1,
  "provinciaNombre": "Buenos Aires",
  "localidadId": 147,
  "localidadNombre": "Saladillo",
  "cantidad": 1
}
```

PROCESO INTERNO:

1. **Login** → Cookies: `.AspNetCore.Cookies`, `ARRAffinity`, etc.
2. **Cotización BCRA** → 1455 (ejemplo)
3. **Construir XML** → `<df><config>...</config><dataset>...</dataset></df>`
4. **POST Xubio** → Response XML con TransaccionID
5. **Parsear** → Extraer datos útiles

OUTPUT (lo que recibís):

```
{
  "success": true,
  "message": "Factura creada exitosamente",
  "data": {
    "transaccionId": "67750488",
    "numeroDocumento": "A-00004-00001680",
    "total": 593.9,
    "pdfUrl": "https://xubio.com/NXV/transaccion/ver/67750488",
    "cotizacion": 1455,
    "cantidad": 1
  }
}
```

🔗 INTEGRACIÓN CON APPS SCRIPT

Una vez validado el endpoint, modificar Apps Script:

Código para Apps Script

```
/**
 * NUEVA FUNCIÓN - Usar endpoint Vercel
 * Reemplaza la función OAuth anterior
 */
```

```
function crearFacturaViaVercel(cliente, cantidad = 1) {
  // CAMBIAR ESTA URL por la real de Vercel
  const VERCEL_ENDPOINT = 'https://TU-PROYECTO.vercel.app/api/crear-factura';

  const payload = {
    clienteId: cliente.id,
    clienteNombre: cliente.nombre,
    provinciaId: cliente.provinciaId,
    provinciaNombre: cliente.provinciaNombre,
    localidadId: cliente.localidadId,
    localidadNombre: cliente.localidadNombre,
    cantidad: cantidad
  };

  const options = {
    method: 'post',
    contentType: 'application/json',
    payload: JSON.stringify(payload),
    muteHttpExceptions: true
  };

  Logger.log('📡 Llamando a endpoint Vercel...');

  const response = UrlFetchApp.fetch(VERCEL_ENDPOINT, options);
  const responseCode = response.getResponseCode();
  const responseText = response.getContentText();

  Logger.log('📡 Response Code: ' + responseCode);

  if (responseCode !== 200) {
    Logger.log('❌ Error: ' + responseText);
    throw new Error('Error al crear factura: ' + responseText);
  }

  const resultado = JSON.parse(responseText);

  if (!resultado.success) {
    throw new Error('Error: ' + resultado.error);
  }

  Logger.log('✅ Factura creada exitosamente');
  Logger.log('TransaccionID: ' + resultado.data.transaccionId);
  Logger.log('Número: ' + resultado.data.numeroDocumento);
  Logger.log('PDF: ' + resultado.data.pdfUrl);

  return resultado.data;
}

/**
 * TEST con endpoint Vercel
 */
function testCrearFacturaVercel() {
  const cliente = {
    id: 8157173,
```

```
    nombre: '2MCAMPO',
    provinciaId: 1,
    provinciaNombre: 'Buenos Aires',
    localidadId: 147,
    localidadNombre: 'Saladillo'
  };

  const resultado = crearFacturaViaVercel(cliente, 1);

  return resultado;
}
```

Pasos:

1. Copiar código arriba a Apps Script
2. Cambiar `TU-PROYECTO.vercel.app` por URL real
3. Ejecutar `testCrearFacturaVercel()`
4. Ver logs (Ctrl+Enter)

TROUBLESHOOTING

Error: "Missing credentials"

Causa: Variables de entorno no configuradas o no visibles **Solución:**

1. Verificar que `XUBIO_USERNAME` y `XUBIO_PASSWORD` existen en Vercel Settings
2. Verificar que están marcadas para `Production`, `Preview`, `Development`
3. Hacer **Redeploy** obligatorio
4. Esperar que redeploy termine completamente

Error: "Login falló - No se redirigió a xubio.com"

Causa: Credenciales incorrectas o Visma Connect bloqueó **Solución:**

1. Verificar credenciales con login manual en `https://xubio.com`
2. Password tiene comillas (`Corvus"22`) - verificar que esté correcto
3. Ver logs de Vercel: `vercel logs --follow`

Error: "Function timeout"

Causa: Playwright tardó más de 60 segundos **Solución:**

1. Verificar `vercel.json` tiene `maxDuration: 60`
2. Puede ser cold start (primera ejecución) - reintentar
3. Ver logs de Vercel para entender dónde se traba

Error: "Error de Xubio: NullPointerException"

Causa: XML incompleto o datos inválidos **Solución:**

- 1. Verificar que `clienteId`, `provinciaId`, `localidadId` son válidos en Xubio
- 2. Ver logs de Vercel para ver XML generado
- 3. Comparar con template GOLD en `docs/Consulta APIs/TEMPLATE_GOLD_XML_LEGACY.xml`

Ver Logs de Vercel

Opción A - Dashboard:

- 1. Vercel Dashboard → Tu proyecto → Deployments
- 2. Click en el deployment activo
- 3. Tab "Functions"
- 4. Click en función que falló
- 5. Ver logs completos

Opción B - CLI:

```
vercel login
vercel logs --follow
```

☑ VALIDACIÓN FINAL

Checklist de Éxito (Railway o Vercel)

- ☐ Deploy completo en la plataforma elegida (Railway o Vercel)
- ☐ Variables `XUBIO_USERNAME` y `XUBIO_PASSWORD` configuradas
- ☐ Redeploy/Build completado exitosamente
- ☐ Test `/api/test-login` retorna `{"success": true}`
- ☐ Test `/api/crear-factura` retorna `TransaccionID` válido
- ☐ Factura visible en Xubio web
- ☐ PDF accesible desde `pdfUrl` de la response
- ☐ Apps Script puede llamar al endpoint exitosamente

Comparación de Plataformas

Feature	Railway.app (Free)	Vercel (Free)	Vercel (Hobby \$20/mes)
Timeout	☑ Sin límites	✗ 10s (insuficiente)	☑ 60s
Memory	☑ Suficiente	⚠ 1024MB (justo)	☑ 3008MB
Costo	\$5 crédito/mes	Gratis	\$20/mes
Playwright	☑ Funciona	✗ Hace timeout	☑ Funciona
Setup	☑ Simple	☑ Simple	☑ Simple
Recomendado	☑ Sí	✗ NO	☑ Sí

- **Guía completa:** [docs/VERCEL_ENDPOINT_README.md](#)
 - **Variables de entorno:** [docs/ENV_VARS.md](#)
 - **Template XML GOLD:** [docs/Consulta APIs/TEMPLATE_GOLD_XML_LEGACY.xml](#)
 - **Plan original:** [docs/planes/plan-endpoint-vercel-xubio.md](#)
-

RESUMEN EJECUTIVO

¿Qué hace este endpoint?

Automatiza la creación de facturas en Xubio sin depender de OAuth (que no funciona). Usa Playwright para hacer login programático, obtener cookies de sesión y enviar facturas vía XML Legacy.

¿Cuánto tarda?

- Primera vez (cold start): ~15 segundos
- Llamadas posteriores: ~10-12 segundos
- Con cache (Fase 5 opcional): ~2-3 segundos

¿Es confiable?

Sí, basado en template GOLD validado en producción. Playwright es robusto para browser automation.

¿Qué pasa si falla?

Logs detallados en Vercel permiten debugging. Errores comunes documentados en Troubleshooting.

Última actualización: 2025-12-31 22:30 UTC-3 **Estado:** ☒ Listo para deployment en Railway.app o Vercel

Próximo checkpoint: Elegir plataforma y deployar **Plataforma recomendada:** Railway.app (free tier funciona perfectamente)