



Universidad Nacional de Mar del Plata

FACULTAD DE INGENIERÍA

TRABAJO FINAL: INGENIERÍA  
ELECTRÓNICA

*Desarrollo de Analizador Vectorial de Redes  
para banda de UHF*

Vazquez, Leonardo David

Noviembre 2024



RINFI es desarrollado por la Biblioteca de la Facultad de Ingeniería de la  
Universidad Nacional de Mar del Plata.

Tiene como objetivo recopilar, organizar, gestionar, difundir y preservar  
documentos digitales en Ingeniería, Ciencia y Tecnología de Materiales y  
Ciencias Afines.

A través del Acceso Abierto, se pretende aumentar la visibilidad y el impacto  
de los resultados de la investigación, asumiendo las políticas y cumpliendo  
con los protocolos y estándares internacionales para la interoperabilidad  
entre repositorios



Esta obra está bajo una [Licencia Creative Commons  
Atribución- NoComercial-CompartirIgual 4.0  
Internacional](https://creativecommons.org/licenses/by-nc-sa/4.0/).



Universidad Nacional de Mar del Plata

FACULTAD DE INGENIERÍA

TRABAJO FINAL: INGENIERÍA  
ELECTRÓNICA

*Desarrollo de Analizador Vectorial de Redes  
para banda de UHF*

Vazquez, Leonardo David

Noviembre 2024

## Dedicatoria

Este proyecto simboliza una de las etapas más valiosas de mi vida y de mi carrera profesional. Por ello, lo dedico con gratitud a mi familia y a mis amigos, quienes siempre me han apoyado y escuchado, tanto en los momentos más difíciles como en los de mayor alegría. Y, por supuesto, a la memoria de mi abuelo, quien me legó la curiosidad por la electrónica.

## Agradecimientos

Con profundo respeto y reconocimiento, expreso mi más sincera gratitud a mis directores de Trabajo Final, el Dr. Ing. Ramiro Abalos Ribas y el Ing. Brian Gluzman, así como a mis directores de Beca de Investigación, el Dr. Ing. Alejandro Uriz y el Dr. Ing. Jorge Castiñeira. Ellos estuvieron presentes en cada etapa del proyecto, brindándome su apoyo, orientación y consejo en todos los aspectos.

Extiendo mi agradecimiento a mis colegas del Laboratorio de Comunicaciones y del Departamento de Ingeniería Electrónica y Computación, tanto docentes como estudiantes, por su generoso apoyo y tiempo, que resultaron fundamentales en mi camino de formación académica y de investigación.

Finalmente, expreso mi reconocimiento a la Universidad Nacional de Mar del Plata, cuya educación de calidad ha fomentado en mí un profundo sentimiento de pertenencia invaluable. Es un honor formar parte de una institución de tal magnitud.

# Índice

<b>1. Introducción</b>	<b>6</b>
1.1. Contexto . . . . .	6
1.2. Partes interesadas . . . . .	6
1.3. Estructura del informe . . . . .	6
<b>2. Fundamentos y estado del arte</b>	<b>8</b>
2.1. Analizador Vectorial de Redes . . . . .	8
2.2. Parámetros S . . . . .	8
2.3. Acopladores direccionales . . . . .	10
<b>3. Descripción del sistema</b>	<b>12</b>
3.1. Alcance del proyecto . . . . .	12
3.2. Diagrama del sistema . . . . .	12
3.2.1. Circuito electrónico (Hardware) . . . . .	13
3.2.2. Programa ejecutable (Software) . . . . .	14
3.3. Especificaciones funcionales . . . . .	15
3.3.1. RF01: Configuración del rango de medición . . . . .	15
3.3.2. RF02: Barrido en frecuencia . . . . .	15
3.3.3. RF03: Generación y detección de señales . . . . .	15
3.3.4. RF04: Procesamiento y envío de datos . . . . .	15
3.3.5. RF05: Muestra de resultados . . . . .	15
3.3.6. RF06: Calibración . . . . .	15
<b>4. Placa de circuitos electrónicos: Hardware</b>	<b>16</b>
4.1. Microcontrolador programador . . . . .	17
4.1.1. Esquemático y PCB . . . . .	17
4.1.2. Programa del PIC (Firmware) . . . . .	19
4.2. Generadores de RF . . . . .	20
4.2.1. Registros . . . . .	20
4.2.2. Esquemáticos . . . . .	22
4.2.3. PCB . . . . .	26
4.3. Acoplador direccional resistivo . . . . .	27
4.3.1. Esquemático . . . . .	31
4.3.2. Diseño en PCB . . . . .	31
4.4. Receptor heterodino . . . . .	33
4.4.1. Multiplicadores . . . . .	33
4.4.2. Divisor de potencia . . . . .	34
4.4.3. Esquemáticos . . . . .	35
4.4.4. Diseño de PCB . . . . .	36
4.5. Adquisición . . . . .	37
4.5.1. CODECs de audio . . . . .	37
4.5.2. Restador . . . . .	39
4.5.3. Recepción y adquisición . . . . .	39
4.5.4. Esquemáticos . . . . .	40
4.5.5. Diseño en PCB . . . . .	41
4.6. Ecuaciones Finales . . . . .	42
4.6.1. Coeficientes de reflexión S11 y S22 . . . . .	42

4.6.2.	Coeficientes de transmisión S21 y S12 . . . . .	43
4.7.	Placa completa . . . . .	44
4.7.1.	Diseño en PCB . . . . .	44
4.7.2.	Placa Final . . . . .	44
4.7.3.	Ficha Técnica . . . . .	47
4.7.4.	Características eléctricas . . . . .	47
4.7.5.	Características de RF . . . . .	47
4.7.6.	Conectores e Interfaces . . . . .	47
<b>5.</b>	<b>Programa ejecutable: Software</b>	<b>48</b>
5.1.	Interfaz de usuario (Frontend) . . . . .	49
5.2.	Paneles y módulos (Backend) . . . . .	50
5.2.1.	Menu Tab . . . . .	50
5.2.2.	Panel de frecuencia . . . . .	50
5.2.3.	Panel de conexión USB . . . . .	51
5.2.4.	Panel de parámetros S . . . . .	52
5.2.5.	Panel de calibración . . . . .	52
5.3.	Algoritmos principales . . . . .	54
5.3.1.	Barrido de frecuencias (Controlador) . . . . .	54
5.3.2.	Envío de frecuencias (SEND UART) . . . . .	55
5.3.3.	Procesamiento de señales (DSP) . . . . .	55
5.4.	Calibración . . . . .	57
5.4.1.	Calibración en reflexión (GAMMA CORRECTOR) . . . . .	57
5.4.2.	Calibración en transmisión (TRANSFER CORRECTOR) . . . . .	58
<b>6.</b>	<b>Plan de pruebas</b>	<b>60</b>
6.1.	Ambientes de prueba . . . . .	60
6.2.	Recursos, herramientas e instrumentos . . . . .	61
6.3.	Políticas de trabajo . . . . .	61
6.4.	Pruebas de verificación . . . . .	61
6.4.1.	Desarrollo de Firmware en PIC . . . . .	62
6.4.2.	Generación de señales de RF . . . . .	62
6.4.3.	Recepción de señales de RF . . . . .	64
6.4.4.	Adquisición de señales . . . . .	65
6.4.5.	Barrido de frecuencias . . . . .	66
6.4.6.	Procesamiento de señales y parámetros S . . . . .	67
<b>7.</b>	<b>Mediciones y resultados</b>	<b>68</b>
7.1.	Pruebas de calibración . . . . .	68
7.1.1.	Reflexión . . . . .	68
7.1.2.	Transmisión . . . . .	70
7.2.	Mediciones DUT . . . . .	71
7.2.1.	Resonador 2 GHz . . . . .	71
7.2.2.	Atenuador 10dB . . . . .	73
7.2.3.	Filtro 1,8 GHz . . . . .	73
7.3.	Interpretación de resultados . . . . .	74
<b>8.</b>	<b>Conclusiones y trabajo futuro</b>	<b>75</b>
8.1.	Logros principales . . . . .	75
8.2.	Recomendaciones y trabajo futuro . . . . .	75

<b>9. Anexos</b>	<b>77</b>
9.1. Diseño de PCB en RF	77
9.1.1. Pérdidas y efectos indeseados	77
9.1.2. Compatibilidad Electromagnética	78
9.2. Acoplador resistivo direccional	81
9.2.1. Marco Teórico	82
9.2.2. Diagrama final	89
9.2.3. Hipótesis	94
9.2.4. Implementación en VNA	95
9.2.5. Simulaciones con método de elementos finitos	96
9.2.6. Parámetros S	97
9.3. Esquemáticos	102
9.4. Lista de componentes	115
9.5. Firmware en Embedded C	118
9.5.1. Librerías	118
9.5.2. Configuración	118
9.5.3. Definiciones	118
9.5.4. Función de parpadeo de LEDs	118
9.5.5. Función de carga de registros	119
9.5.6. Funciones UART	119
9.5.7. Cálculo de Registros	120
9.5.8. Programa principal: Main	122
9.5.9. Funciones de inicio	122
9.6. Algoritmos principales en Python	124
9.6.1. Librerías	124
9.6.2. Envío de frecuencias por UART	124
9.6.3. Adquisición de señales	124
9.6.4. Filtro digital pasabanda	125
9.6.5. Cálculo de amplitud y fase mediante FFT	125
9.6.6. Grabación y procesamiento de señales	126
9.6.7. Cálculo de Registros	126
9.6.8. Función de barrido en frecuencia	128
9.6.9. Error en reflexión y correctores	129
9.6.10. Funciones gráficas	129
9.6.11. Mediciones S21/S12	130
9.6.12. Mediciones S11/S22	131
9.6.13. Función de Suavizado Savitsky-Golay	131

## Definiciones, siglas y abreviaturas

Nombre	Descripción
VNA	<i>Vectorial Network Analyzer</i>
LAC	Laboratorio de Comunicaciones
ICYTE	Instituto de Ciencia y Tecnología en Electrónica
UNMDP	Universidad Nacional de Mar del Plata
RF	Radio Frecuencias
DUT	<i>Device Under Test</i>
PC	Computador Personal
PCB	<i>Printed Circuit Board</i>
SMT	<i>Surface Mount Technology</i>
MMIC	<i>Monolithic Microwave Integrated Circuits</i>
UHF	<i>Ultra High Frequency</i>
USB	<i>Universal Serial Bus</i>
IF	<i>Intermediate Frequency</i>
UART	<i>Universal Asynchronous Receiver-Transmitter</i>
ADC	<i>Analog-to-Digital Converter</i>
FR4	<i>Flame Retardant 4 Material</i>
SPI	<i>Serial Peripheral Interface</i>
LED	<i>Light-Emitting Diode</i>
PLL	<i>Phase-Locked Loop</i>
CODEC	<i>Encoder-Decoder</i>
THR	<i>Through</i>
WAV	<i>Waveform audio file format</i>
FFT	<i>Fast Fourier Transform</i>
ICSP	<i>In Circuit Serial Programming</i>

Tabla 1: Definiciones, siglas y abreviaturas.



# 1. Introducción

El presente documento trata el desarrollo de un analizador vectorial de redes (*Vector network analyzer*, o simplemente VNA), el cual es un instrumento de medición para la caracterización de diversos sistemas, ya sean cables, amplificadores, sensores y/o antenas en el rango de frecuencias de microondas. Este aparato es capaz de entregar parámetros que son de suma importancia en el área de las comunicaciones.

## 1.1. Contexto

Desde el Laboratorio de Comunicaciones (LAC) del Instituto de Investigaciones Científicas y Tecnológicas en Electrónica (ICYTE) se requiere la construcción de un dispositivo analizador vectorial de redes. La idea de este aparato surge a partir de una línea de investigación del laboratorio: El desarrollo de un sistema de medición no invasiva de glucemia, cuya optimización se lograría precisamente con un analizador vectorial de redes.

Cabe destacar que el integrante de este trabajo final pertenece al grupo de investigación mencionado bajo una beca de grado de la Universidad Nacional de Mar del Plata (UNMDP). Por esta razón, la solución no será únicamente la construcción del aparato sino también el valor científico que pueda aportar al proyecto.

## 1.2. Partes interesadas

El grupo de investigación del Laboratorio de Comunicaciones es quien solicita el dispositivo de interés. Ellos serán los clientes y beneficiarios, y quienes se encarguen de la supervisión del proyecto. En particular, el equipo utilizará este dispositivo como una optimización de su medidor no invasivo de glucemia, por lo que su usuario final objetivo son las personas que sufren de la enfermedad de la diabetes.

Por otro lado, este trabajo está destinado para los siguientes usuarios:

- Técnicos, ingenieros y/o radioaficionados que utilicen esta herramienta como instrumento de medición para sus trabajos.
- Estudiantes interesados en utilizar el dispositivo para mejorarlo o para sus trabajos propios.
- Laboratorio de Comunicaciones.

## 1.3. Estructura del informe

Este documento está organizado de la siguiente manera:

En primer lugar, la sección 1 brinda una introducción al proyecto y al informe. La sección 2 desarrolla los fundamentos necesarios para entender el funcionamiento del analizador vectorial de redes, en general.

A partir de la sección 3 se empieza a describir los requisitos y las funcionalidades que debe cumplir el sistema. A partir de aquí, en las secciones 4 y 5 se brinda información detallada acerca de la construcción del aparato y el desarrollo del sistema a nivel Software. Para corroborar el funcionamiento del sistema, en la sección 6 se detalla el plan de pruebas para la validación y homologación del mismo, además de mediciones típicas de calibración.

Ya en la sección 7 se muestran algunos resultados comparados con un instrumento de medición profesional, dando paso a la sección 8 de conclusiones y trabajo futuro.

Por otro lado, en el Anexo se encuentra información de:

- Reglas de diseño en RF.
- Acoplador direccional resistivo.
- Esquemáticos.
- Componentes.
- Código Firmware.
- Código Backend.

Finalmente se encuentran las referencias y la bibliografía citada.

## 2. Fundamentos y estado del arte

### 2.1. Analizador Vectorial de Redes

Un analizador vectorial de redes (VNA) básico posee tanto una fuente de radio frecuencias (RF), utilizada para generar una señal de estímulo conocida, como un conjunto de receptores, los cuales son utilizados para determinar los cambios en este estímulo causados por el dispositivo bajo prueba o DUT (Device under test).

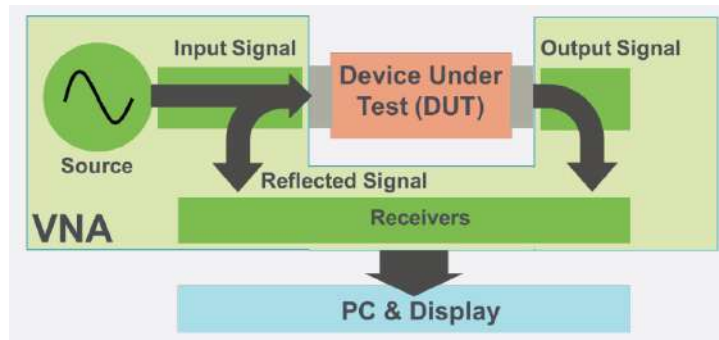


Figura 1: Esquema básico del Analizador Vectorial de Redes. Fuente: [1].

La señal de estímulo o incidente se inyecta en el DUT y el VNA mide tanto la señal que se refleja desde el lado de entrada como la señal que pasa hacia el lado de salida del DUT. Los receptores del VNA miden las señales resultantes y las comparan con la señal de estímulo conocida. Los resultados medidos son procesados por una PC y enviados a una pantalla o interfaz de usuario.

### 2.2. Parámetros S

Los parámetros S (*Scattering parameters*) se utilizan para caracterizar las propiedades eléctricas de un sistema o dispositivo de RF. Para comprender cómo utilizar un VNA para caracterizar un DUT, es importante entender los conceptos básicos de los parámetros S.

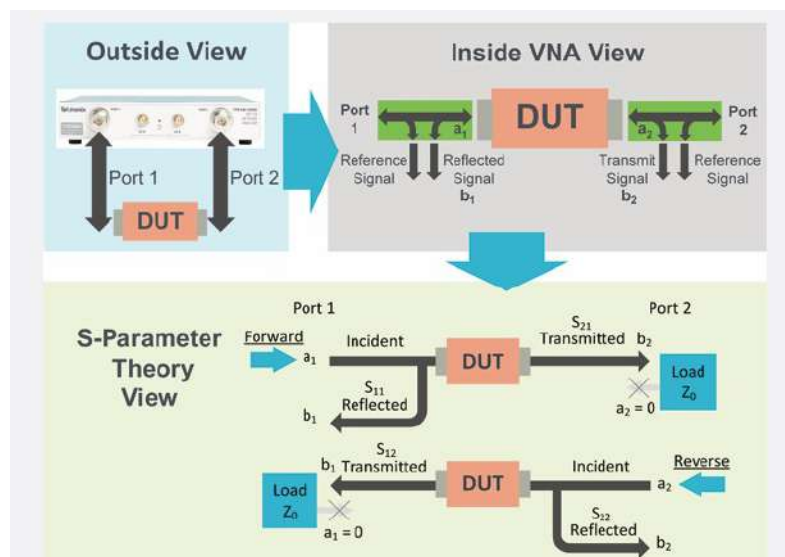


Figura 2: Medición de parámetros S con VNA. Fuente: [1].

Un VNA típicamente tiene dos o más puertos que se conectan al DUT. Estos puertos están etiquetados, en este caso, como Puerto 1 y Puerto 2. En general, la señal que se inyecta al DUT

se denomina onda incidente y las que salen se denominan onda reflejada y onda transmitida. En primer lugar, se toma una porción de la señal inyectada como señal de referencia de la onda incidente. Al mismo tiempo, parte de dicha señal se refleja al entrar en el DUT y se separa con el acoplador, siendo esta la onda reflejada. Por el otro lado, la porción de la señal que incide al DUT generalmente experimenta cambios en magnitud y fase a medida que pasa debido a su transferencia, y sale hacia el Puerto 2 como onda transmitida. Los parámetros S se definen como razones o cocientes de las señales que provienen de varios puertos en relación con la señal referencia (figura 3).

	Forward:	Reverse:
Reflection:	$S_{11} = \frac{\text{Reflected}}{\text{Incident}} = \frac{b_1}{a_1} \Big _{a_2=0}$	$S_{22} = \frac{\text{Reflected}}{\text{Incident}} = \frac{b_2}{a_2} \Big _{a_1=0}$
Transmission:	$S_{21} = \frac{\text{Transmitted}}{\text{Incident}} = \frac{b_2}{a_1} \Big _{a_2=0}$	$S_{12} = \frac{\text{Transmitted}}{\text{Incident}} = \frac{b_1}{a_2} \Big _{a_1=0}$

Figura 3: Definición de parámetros S. Fuente: [1].

Hay una variedad de VNAs disponibles en el mercado, cada uno con un número diferente de puertos y caminos por los que fluye la señal de estímulo. En el caso de un VNA de 1 puerto, el DUT está conectado al lado de entrada de la figura 2 y solo se pueden medir las señales reflejadas. Para un VNA de 2 puertos y 1 camino, se pueden medir tanto la señal reflejada como la transmitida (parámetros S11 y S21), sin embargo, el DUT debe ser invertido físicamente para medir los parámetros inversos (parámetros S22 y S12). En cuanto a un VNA de 2 puertos y 2 caminos, el DUT puede conectarse a cualquier puerto en cualquier dirección porque el instrumento tiene la capacidad de invertir el flujo de señal para que se puedan medir las reflexiones en ambos puertos (parámetros S11 y S22), así como las transmisiones hacia adelante y hacia atrás (S21 y S12).

En la figura 4 se observa un analizador vectorial de redes comercial de 2 puertos con un ancho de banda de 3 GHz:



Figura 4: analizador vectorial de redes RIGOL RSA3030N. Fuente: [2].

### 2.3. Acopladores direccionales

Una de las métricas más importantes para indicar la performance de cualquier sistema o red de RF es cuán bien adaptado está en términos de impedancia, brindando así información acerca de la transmisión de potencia y las pérdidas asociadas. Esta caracterización requiere la discriminación entre las señales incidente y reflejada. Los dispositivos que realizan esta función son los acopladores direccionales.

Los acopladores direccionales son circuitos de cuatro puertos de redes pasivas cuyo propósito en reflectometría es tomar muestras de las señales incidente y reflejada. En la figura siguiente se observa un diagrama de un bloque básico de un acoplador bidireccional, en la cual el puerto 1 es el puerto incidente, el puerto 2 es el puerto pasante (donde comúnmente se coloca la carga del sistema a medir), el puerto 3 es el puerto acoplado al puerto 2 y aislado al puerto 1, mientras que el puerto 4 es el puerto aislado al puerto 2 pero acoplado al puerto 1.

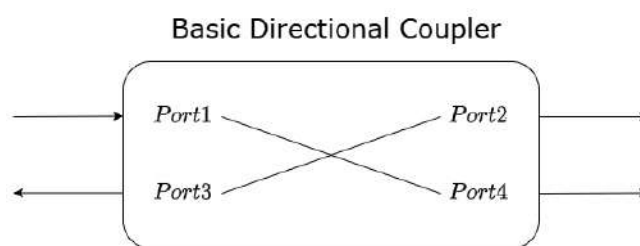


Figura 5: Acoplador Direccional básico.

Las siguientes expresiones indican las definiciones matemáticas que se emplean en ingeniería de microondas para caracterizar este tipo de sistemas [3].

Pérdida de retorno (*Return loss*)

$$R_L = -20 \cdot \log(|S_{11}|) \quad (1)$$

Pérdida de inserción (*Insertion loss*)

$$I_L = -20 \cdot \log(|S_{21}|) \quad (2)$$

Aislamiento (*Isolation*)

$$I = -20 \cdot \log(|S_{41}|) \quad (3)$$

Acoplamiento (*Coupling*)

$$C = -20 \cdot \log(|S_{31}|) \quad (4)$$

Directividad (*Directivity*)

$$D = 20 \cdot \log(|S_{31}/S_{41}|) \quad (5)$$

En el mercado existen diversas tecnologías que permiten sintetizar el funcionamiento de un acoplador direccional. Los acopladores direccionales más comunes emplean tecnologías de parámetros distribuidos como Microstrip [4], en el cual se aprovecha el acoplamiento entre líneas de transmisión. El problema principal de este tipo de acopladores es que depende de la longitud de onda de la frecuencia central a la cual está diseñado, por lo que posee un ancho de banda acotado. Aún así, son muy sencillos de diseñar a implementar en una placa de circuitos impresos (PCB).

Por otro lado, y debido al avance de tecnología de montaje superficial (SMT), han aparecido acopladores basados en puentes direccionales resistivos incluso embebidos en chips MMIC como el ADL5920 ([5]), que presentan una característica de ancho de banda considerable logrando así

medidas de directividad similares a los acopladores clásicos a costa de mayor pérdida de inserción. Estos arreglos emplean componentes de parámetros concentrados, logrando que el tamaño del dispositivo sea pequeño en términos de longitud de onda, logrando así disminuir las pérdidas generadas por el efecto skin y disminuyendo la diferencia de caminos en términos de fase de las señales.



Figura 6: Comparación en tamaño entre un acoplador convencional, el kit de evaluación del ADL5920 (VWRS-meter) y el chip integrado ADL5920. Fuente: [5].

### 3. Descripción del sistema

Este proyecto propone el diseño, desarrollo, calibración y ensayo de un sistema analizador vectorial de redes para la medición de parámetros S de sistemas de comunicaciones de radiofrecuencia (RF).

#### 3.1. Alcance del proyecto

Se requiere que el sistema opere en una banda de frecuencias UHF (Ultra high frequency), que va desde 1800 MHz a 2000 MHz. A pesar de este requisito, se centra el desarrollo en toda la banda de UHF, que va desde 300 MHz a 3 GHz. Para ello se emplean dispositivos de montaje superficial junto a circuitos integrados comerciales que operan en UHF. Para las mediciones pertinentes se utiliza instrumental de laboratorio, ofrecido por el Laboratorio de Comunicaciones (LAC-ICYTE-UNMDP).

En resumen, las especificaciones generales que debe cumplir el aparato son:

- Obtener parámetros S de sistemas y circuitos electrónicos de RF.
- Permitir que el usuario pueda configurar el rango de medición y pueda calibrar el instrumento mediante cargas conocidas estándares.

Mientras que los requerimientos de rendimiento son:

- Rango de medición: 1800 a 2000 MHz (UHF opcional).
- Impedancia característica: 50  $\Omega$ .

#### 3.2. Diagrama del sistema

En la figura 7 se observa el diagrama del analizador propuesto, el cual es de 2 puertos y 1 camino. Para poder medir los parámetros S del DUT, se aplican técnicas basadas en Reflectometría y heterodinación [6] a partir de un bloque de circuitería electrónica (Hardware). Mediante este último, se precisa que sea comandado por un bloque controlador y de procesamiento de señales para el posterior análisis y presentación de la información deseada, mediante una interfaz de usuario en un programa ejecutable de PC (Software).

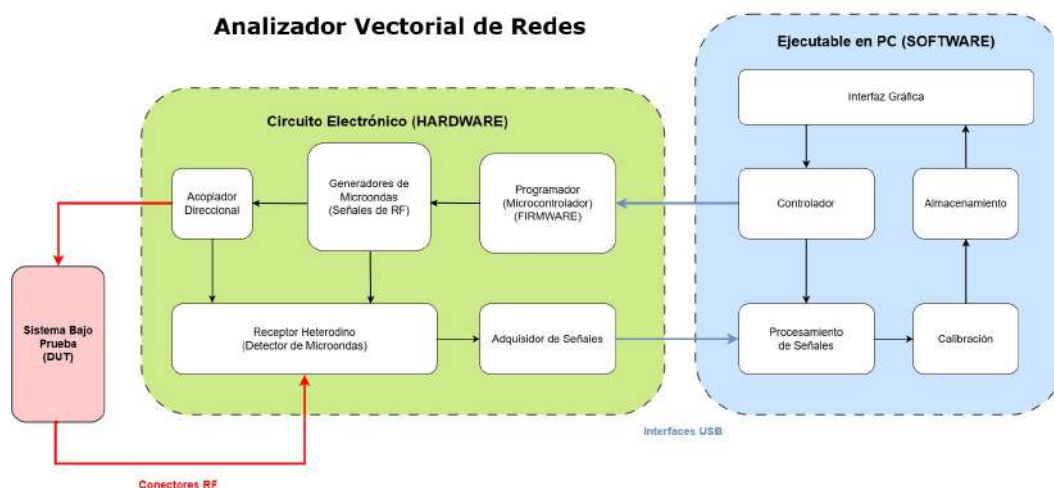


Figura 7: Esquema del Analizador Vectorial de Redes.

En el programa ejecutable el usuario indica al sistema la frecuencia puntual o rango de frecuencias a las cuales se desean medir los parámetros S del DUT. Mediante una conexión USB, el programa le envía los registros de cada frecuencia al microcontrolador programador, el cual en sus líneas de Firmware es capaz de recibir estos datos para luego programar los generadores de RF. Uno de tales generadores es el que se utiliza para inyectar potencia al DUT mientras que el otro se emplea como oscilador local para la etapa de recepción de las señales de RF a medir (Receptor heterodino).

Detrás del Puerto 1 del DUT, hay un acoplador direccional el cual es el encargado de tomar una muestra de la señal incidente y una muestra de la señal reflejada, las cuales van hacia la etapa de recepción de señales de RF junto a la señal de onda transmitida, que es simplemente la conexión del Puerto 2 del DUT.

Las señales de RF correspondientes a las ondas incidente, reflejada y transmitida son llevadas hacia una frecuencia intermedia IF en banda base, específicamente en bandas de audio, para su adquisición y digitalización. En esta etapa se envían por otro puerto USB hacia el programa ejecutable. A partir de aquí se realiza el procesamiento de señal correspondiente para extraer información de magnitudes y fases y así poder calcular el parámetro S correspondiente. El bloque controlador mantiene el orden entre la medición de una frecuencia y la siguiente. Además de esto, en el programa hay una lógica de calibración para corregir errores de conectores y cables utilizados en las mediciones realizadas. Finalmente, en la misma interfaz de usuario se grafican y se guardan los datos deseados.

Se describen el funcionamiento cada una de estas secciones, a nivel Hardware y a nivel Software:

### **3.2.1. Circuito electrónico (Hardware)**

#### **Programador**

Este bloque consta de un microcontrolador cuyo programa (Firmware) recibe los comandos del bloque controlador en Software (a través de una de las interfaces USB) para programar los sintetizadores de radiofrecuencia (RF) del bloque siguiente y asignar un conjunto de frecuencias determinadas.

#### **Generadores de microondas**

Una vez que se programan los sintetizadores, estos deben generar dos señales de radiofrecuencia a una frecuencia determinada para realizar la técnica de heterodinación. Una de tales señales se emplea para inyectar potencia al sistema bajo prueba (DUT), mientras que la otra actúa como un oscilador local para la etapa de detección.

#### **Acoplador direccional**

Este bloque debe ser capaz de tomar una muestra de la señal que incide hacia el DUT y poder separar la señal de onda reflejada debido a este último.

#### **Conectores de RF**

Se emplean conectores de RF para lograr la conexión física entre los puertos del DUT y los puertos del instrumento VNA, el cual consta de un puerto con potencia disponible para inyectar (ZL) y otro puerto para recibir potencia (TX).



## Receptor heterodino

Básicamente el receptor heterodino actúa como un detector de radiofrecuencias. Técnicamente hablando, esta sección consta de un banco de multiplicadores que bajan en frecuencia a las señales de RF incidente, reflejada y transmitida del DUT (mediante la señal de oscilador local del bloque de generadores). La información en alta frecuencia se transfiere a una frecuencia intermedia manejable (más baja) para poder ser acondicionadas para su posterior adquisición.

## Adquisidores de señales

Este bloque se encarga de adquirir y digitalizar a las señales eléctricas con información de incidencia, reflexión y transmisión del DUT para luego enviar los datos hacia el Software mediante las interfaces USB.

## Interfaces USB

Se emplean interfaces USB para producir la comunicación entre la placa desarrollada (Hardware) y el computador personal (PC) en el cual está instalado el programa desarrollado (Software).

### 3.2.2. Programa ejecutable (Software)

#### Procesamiento de señales

Una vez adquiridas las señales eléctricas se digitalizan en datos y se emplean técnicas digitales para obtener información de amplitud y fase de cada una de ellas, para luego calcular mediante algoritmos a los parámetros deseados.

#### Calibración

Este bloque obtiene los datos de amplitud y fase de cada señal para un conjunto a medir (DUT + cables, atenuadores, etc.) y aplica un algoritmo en particular para corregir los errores no deseados pero conocidos propios del instrumento o del banco de medición (cables, atenuadores, etc).

#### Almacenamiento

Este bloque genera un conjunto de datos que corresponden al parámetro S medido a una o varias frecuencias determinadas.

#### Interfaz gráfica

Es el bloque Frontend del Software en el cual el usuario interactúa con el sistema para poder medir y observar mediante gráficos a los parámetros deseados, además de elegir la banda de frecuencias a mostrar y abrir/guardar los datos medidos.

#### Controlador

Es la sección Backend del Software que se encarga específicamente de lograr una sincronización entre las frecuencias de RF a generar y la adquisición de las señales, manteniendo un lazo que logre un barrido frecuencial para un conjunto de mediciones.

### 3.3. Especificaciones funcionales

#### 3.3.1. RF01: Configuración del rango de medición

El usuario debe poder elegir el rango de medición a efectuar por la placa, mediante la interfaz gráfica. La configuración del rango de frecuencias tiene que ser al menos entre 1800 MHz y 2000 MHz, siendo toda la banda UHF una solución optativa.

#### 3.3.2. RF02: Barrido en frecuencia

El bloque controlador debe recibir el rango suministrado por el usuario mediante la interfaz gráfica y luego enviar la frecuencia determinada al programador de los generadores de RF. Además debe mantener la sincronización de la adquisición de las señales.

#### 3.3.3. RF03: Generación y detección de señales

El circuito electrónico tiene que generar las señales de radiofrecuencia para inyectar hacia el DUT y poder detectar las señales de transmisión y reflexión pertinentes. Se emplea la técnica de reflectometría para obtener las señales de onda incidente y reflejada y la técnica de heterodinación para la detección de las señales de RF y su acondicionamiento mediante filtros o amplificadores.

#### 3.3.4. RF04: Procesamiento y envío de datos

A partir del adquisidor en el Hardware y su interfaz USB, se envían las señales digitalizadas hacia el bloque de procesamiento de señales en Software. Este último debe calcular las amplitudes y las fases de cada señal y generar el parámetro S correspondiente a la medición, ya sea reflexión (S11 o S22) o transmisión (S21 o S12), a una frecuencia determinada. Luego se envían los datos hacia el bloque de almacenamiento para mostrarse en la interfaz gráfica o guardarse en un conjunto de archivos determinado.

#### 3.3.5. RF05: Muestra de resultados

El bloque interfaz de usuario debe mostrar los datos obtenidos mediante gráficas de amplitud y fase en función de la banda de frecuencias asignada. Además, para la medición de reflexión, se muestra una gráfica del ábaco de Smith para visualizar la adaptación de los puertos del DUT, referidos al estándar característico de  $50\Omega$ .

#### 3.3.6. RF06: Calibración

Para poder mitigar los errores típicos en la medición de parámetros de reflexión y/o transmisión de un sistema bajo prueba, se emplean técnicas de calibración que logran disminuir el efecto de atenuaciones o ganancias, y/o cambios de fase producidos por el sistema y/o conjunto de conectores y cables para la conexión del DUT con el Hardware del instrumento. Para ello se emplean cargas y conectores estándares en la cual el usuario debe conectar físicamente en los puertos del Hardware, y debe seleccionar en la interfaz gráfica (Software) cada una de las opciones para luego efectuar cada medición y así generar las dos matrices de errores, tanto en el caso de reflexión (S11 o S22) como en el caso de transmisión (S21 o S12).

## 4. Placa de circuitos electrónicos: Hardware

El circuito electrónico que es parte del sistema VNA consta de 5 secciones:

1. Microcontrolador programador y convertor UART/USB 4.1.
2. Generadores de RF (señal incidente y oscilador local) 4.2.
3. Acoplador direccional resistivo (señal incidente y reflejada) 4.3.
4. Receptor heterodino (Banco de Mixers) 4.4.
5. Adquisición en frecuencia intermedia o de audio, puerto USB 4.5.

Para la construcción del mismo se emplea una placa PCB en conjunto con tecnologías de componentes discretos SMT y *Through Hole*, circuitos integrados programables con comunicación UART, circuitos integrados MMIC y circuitos integrados con convertidores ADC y comunicación USB. La placa es construida por un fabricante nacional. Los dispositivos son colocados y soldados en el Laboratorio de Comunicaciones.

La metodología de trabajo empleada para el diseño de Hardware se muestra en la figura 8. El Software empleado para el diseño es Altium Designer [7] (del cual se posee licencia de estudiante). Los programas de simulación empleados son LTSPICE [8], CST [9] (con licencia de investigación) y QUCS [10].

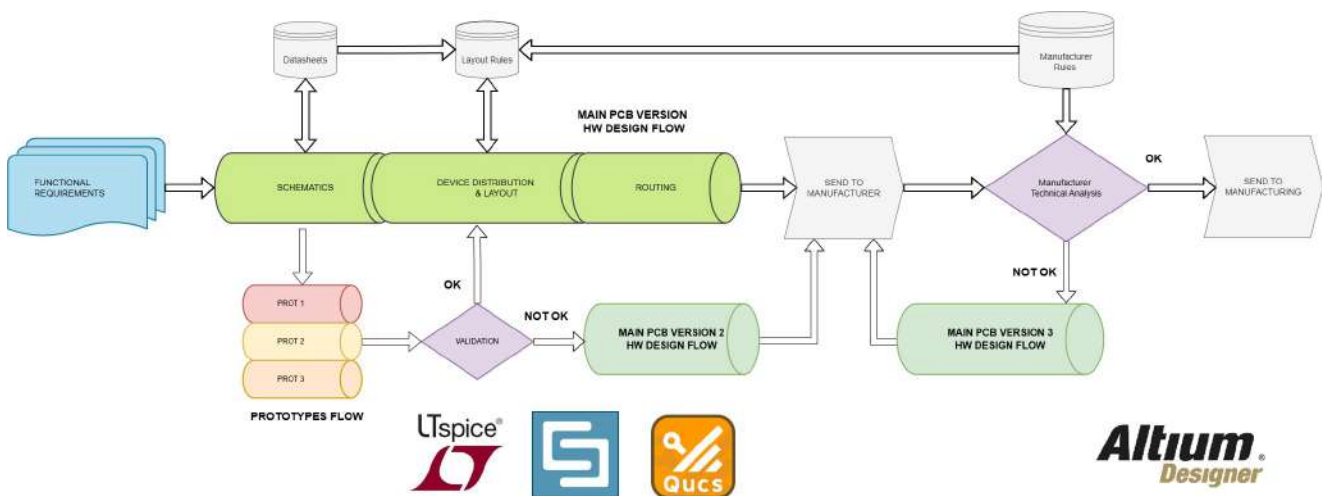


Figura 8: Metodología de Trabajo.

A partir de los requerimientos funcionales, se diseñan los esquemáticos del sistema teniendo en cuenta a los circuitos integrados que cumplen con las características deseadas, a partir de sus respectivas hojas de datos. En paralelo, se construyen prototipos para validar la funcionalidad requerida. En caso de cumplirse, se sigue al flujo de la versión correspondiente para luego comenzar a diseñar el *layout* del PCB, teniendo en cuenta las reglas de cada dispositivo y las de fabricación, las cuales son proveídas por el fabricante de placas. Una vez terminado el diseño y el ruteo de las pistas, se envían los archivos de diseño al fabricante de placas. En caso de que este último valide el diseño, se procede a la construcción y entrega de la placa final.

## 4.1. Microcontrolador programador

Para el programador se emplea un microcontrolador basado en un chip PIC 18F4550 I/PT [11], el cual es el encargado de recibir desde el programa ejecutable en PC a la frecuencia determinada o instantánea a la cual se deben programar los generadores de RF de la etapa siguiente. Para la conexión entre la PC y el PIC se emplea un módulo adaptador conversor MAX232 [12], el cual convierte los niveles de tensión del protocolo USB al protocolo UART RS232 TTL (5 V a 3,3 V).

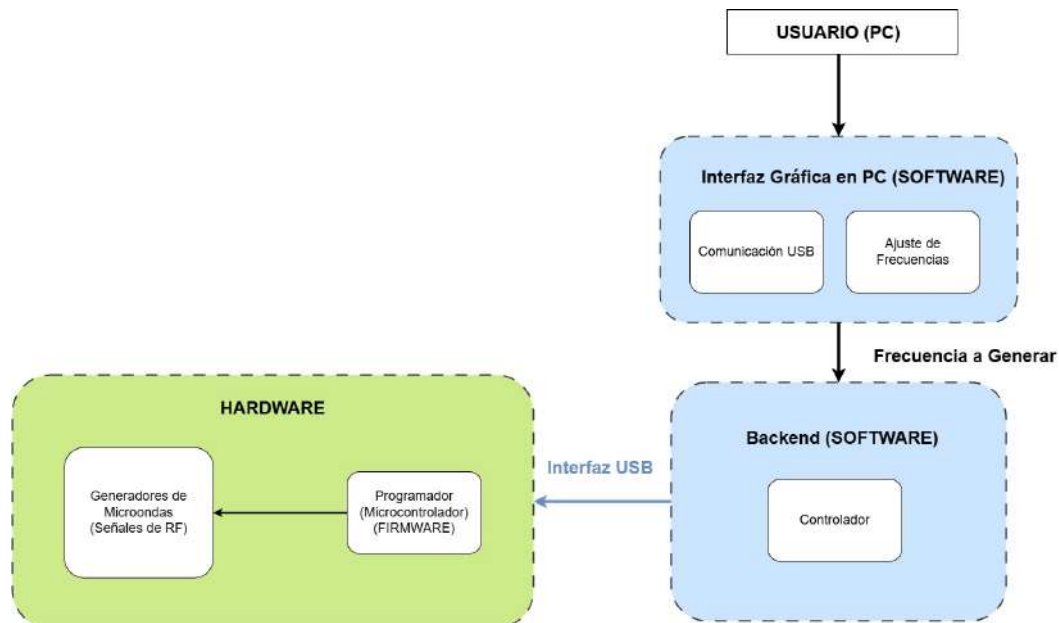


Figura 9: Bloque controlador y programador de frecuencias.

El PIC 18F4550 I/PT es un circuito integrado de la empresa MicroChip que trabaja con 8 bits y un oscilador externo. Entre las características más importantes se destaca:

1. Memoria tipo FLASH. 32 kBytes memoria de programa.
2. 256 Bytes de memoria EEPROM. 2048 Bytes memoria RAM.
3. Puertos de comunicación UART/USART, SPI, I2C.
4. Temporizadores de 16 bits y 8 bits.
5. Periféricos de captura/ comparación/ PWM.
6. Comparadores analógicos.
7. Canales ADC 10 bits.
8. Variedad de interrupciones.

### 4.1.1. Esquemático y PCB

Para la construcción del microcontrolador se emplea una placa PCB de material FR4 doble faz y componentes discretos. La topología del esquemático (figura 10) sigue una característica generalista para emplear el microcontrolador en diferentes aplicaciones, es por eso que se agregan postes hembra de conexión para cables *jumpers* machos, un pulsador de reseteo y algunos LEDs

indicadores. De esta manera, se tiene completa libertad en el uso de los puertos del PIC y sus periféricos. Como oscilador externo se emplea un cristal de cuarzo de 20 MHz en conjunto con capacitores cerámicos SMD de 22pF. Se observa en el diseño de *layout* de la figura 11 que por cada pin se colocan dos postes para facilitar las posibles conexiones. Además, cada línea está separada con un espacio de GND para disminuir el efecto *crosstalk* [13].

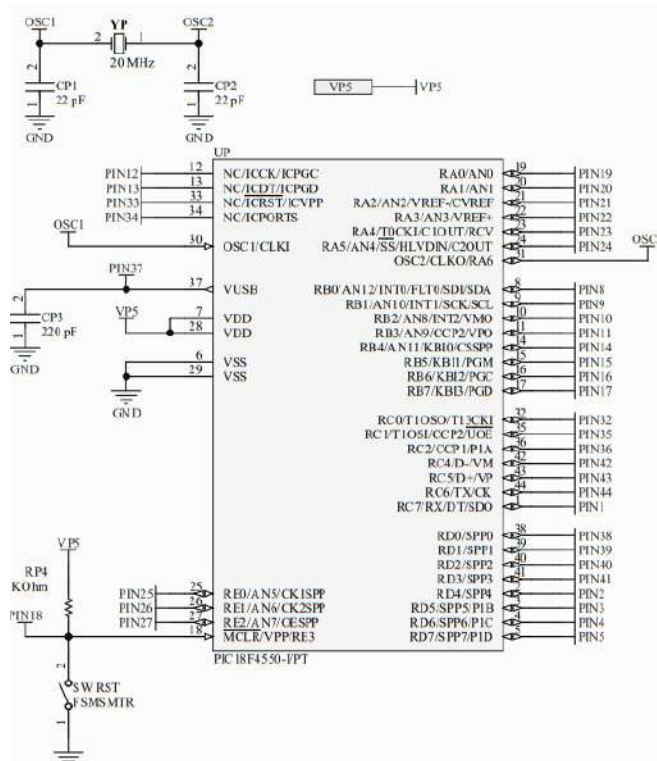


Figura 10: Esquemático del microcontrolador PIC.

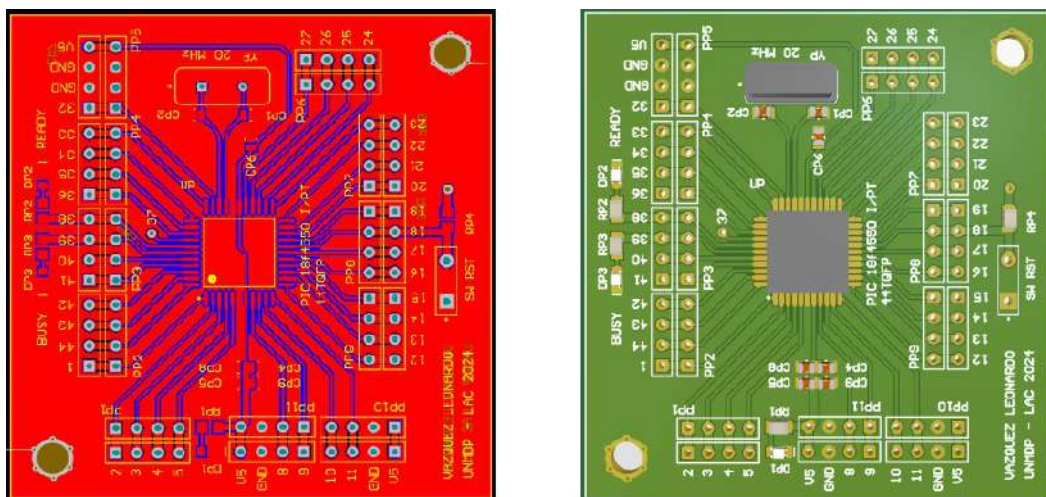


Figura 11: Diseño en *layout* del microcontrolador PIC.

#### 4.1.2. Programa del PIC (Firmware)

El propósito del microcontrolador es programar los registros de los generadores de RF de la siguiente etapa a partir de la frecuencia indicada por el Software en PC. Para ello se programa una serie de acciones al PIC, es decir, el Firmware al cual se ejecutará una vez encendido el microcontrolador. El código utilizado se encuentra en el Anexo 9.

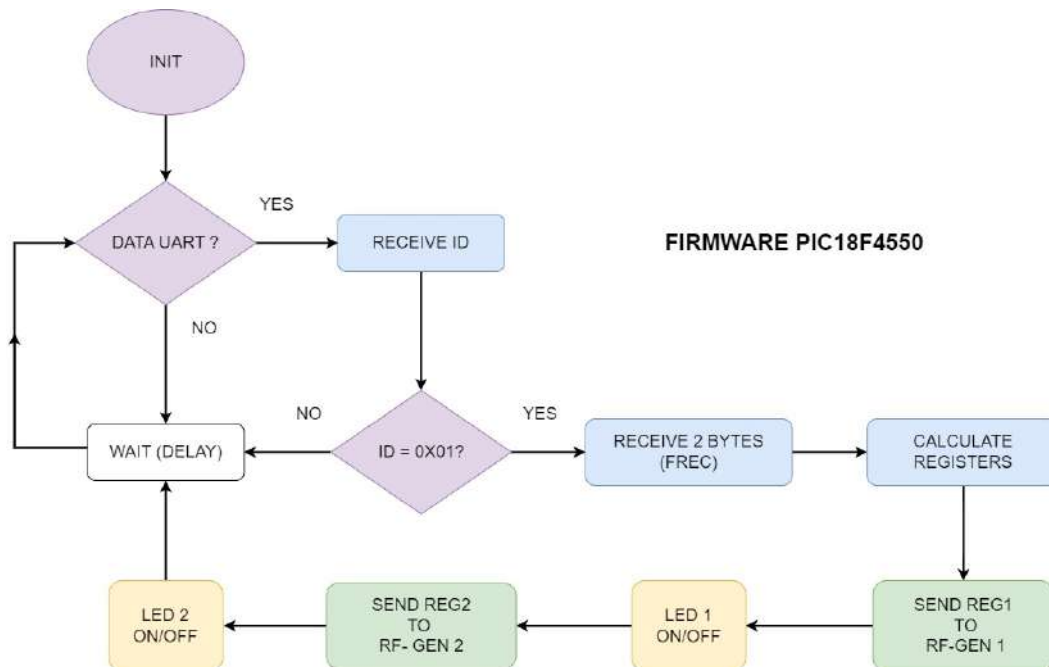


Figura 12: Flujograma del Firmware del microcontrolador PIC.

El flujograma anterior (12) muestra la serie de pasos que realiza el microcontrolador. El funcionamiento es el siguiente: Una vez inicializado o encendido el microcontrolador, se inicializan (INIT) las comunicaciones digitales UART-USB y SPI, para la conexión con la PC y los generadores, a partir de los puertos digitales. Luego, se queda esperando a recibir el dato de frecuencia  $f_o$  proveniente de la PC mediante la interfaz USB-UART.

El protocolo que se emplea para empaquetar la frecuencia consta de una cabecera ID con el payload de 16 bits, es decir, de 2 Bytes, ya que la frecuencia cubre el valor de entero positivo entre 300 y 3000 (MHz). La cabecera le permite al PIC identificar rápidamente el inicio de la transmisión de los datos.

Luego, el PIC calcula los registros correspondientes a cada generador (GEN1 o GEN2), el cual uno de ellos estará programado a una frecuencia  $f_o$ , mientras que el otro a la frecuencia  $f_o + \Delta f$ . De esta manera, se envían los registros mediante el protocolo SPI hacia los generadores, señalizando a través de los parpadeos de LEDS del microcontrolador. Los tiempos de espera o delays se agregan para sincronizar el arribo de datos con el procesamiento de los mismos.

Para programar el PIC y grabar el Firmware en su memoria Flash, se emplea el programa MPLAB [14] para compilar el código y enviarlo, mediante el módulo PICkit V2 [15], hacia el microcontrolador.

## 4.2. Generadores de RF

Los generadores de microondas a emplear como oscilador local y fuente de inyección hacia el DUT están basados en los chips ADF4351 [16] (se emplea un chip por generador), los cuales poseen un arreglo de enganche de fase (PLL) para poder configurar la frecuencia de salida con una resolución de 1 kHz en el rango de frecuencias desde 40 MHz a 4400 MHz, con una potencia disponible de salida máxima de +5dBm. Se observa el diagrama en bloques del chip:

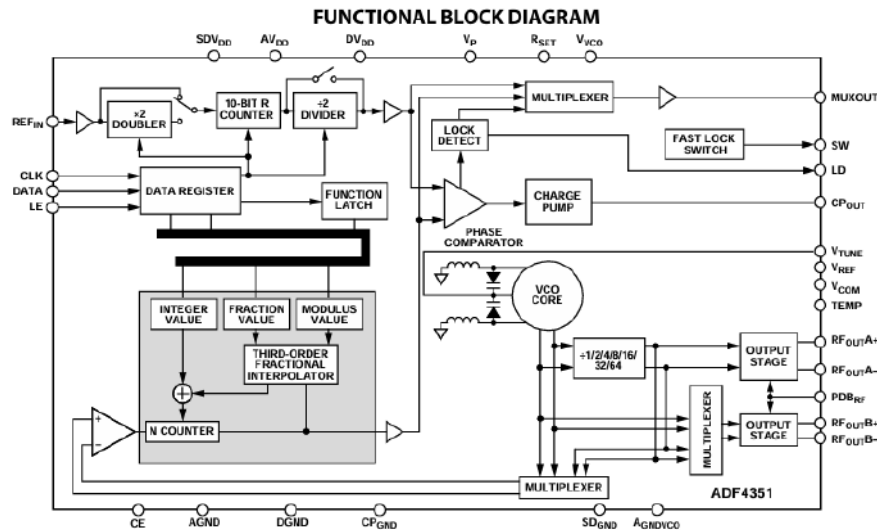


Figura 13: Diagrama en bloques del ADF4351. Fuente: [16].

### 4.2.1. Registros

Internamente, cada chip posee 6 registros los cuales se emplean, entre otras cosas, para configurar la frecuencia y potencia de salida de la señal de RF deseada. Se remarca en la figura 14, en rojo y amarillo, aquellos bits correspondientes a las variables que se utilizan para el cálculo de la frecuencia de salida. En celeste, la fase inicial de la señal, mientras que en azul se habilita la potencia de salida a +5dBm. Es decir que, para una frecuencia y características de señal deseada, existirá un conjunto de valores de los registros los cuales el PIC deberá enviar a cada uno de los generadores mediante la comunicación SPI.

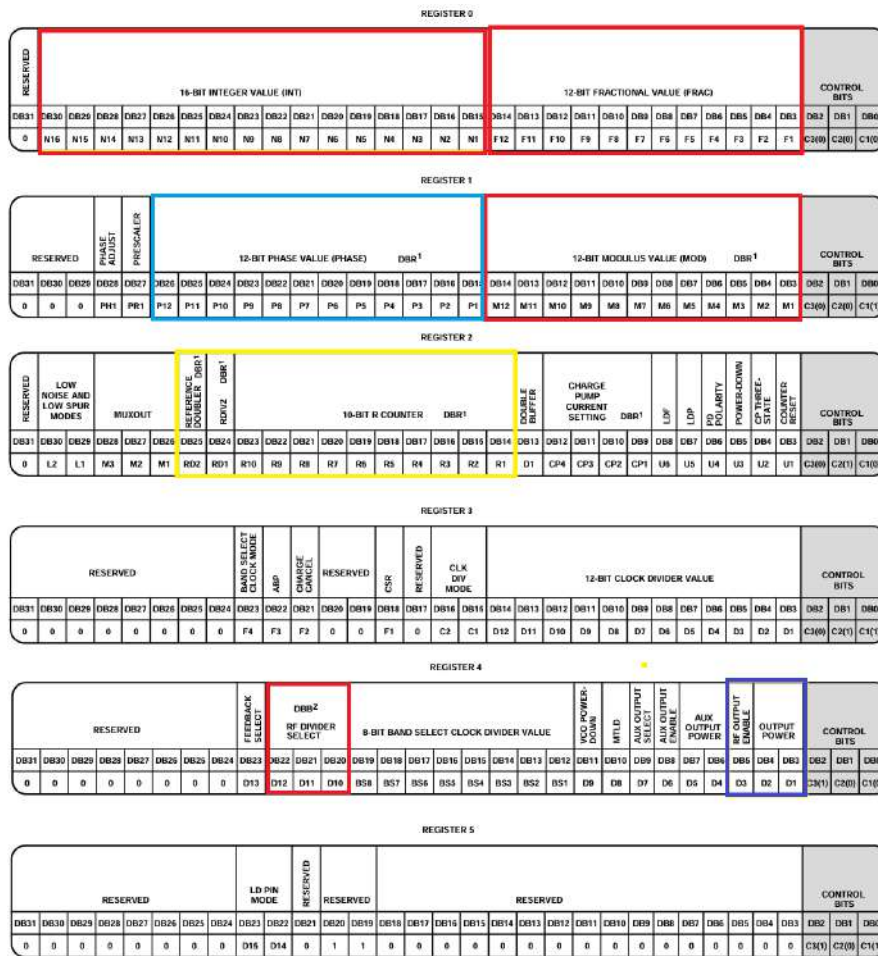


Figura 14: Mapa de bits de cada registro del ADF4351. Fuente: [16].

El funcionamiento del generador está basado en el principio de lazo de enganche de fase (PLL), el cual se emplea un modelo de comparador de fase en conjunto con un oscilador controlado por tensión y un arreglo de divisores y un filtro externo para poder converger la señal a una frecuencia determinada.

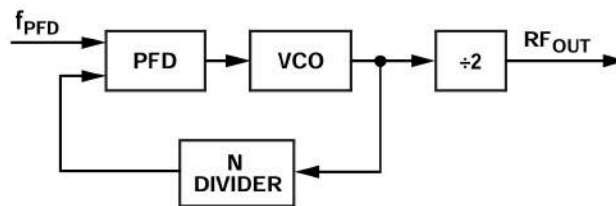


Figura 15: Lazo de enganche de Fase del ADF4351. Fuente: [16].

La frecuencia de salida de un ADF4351 es:

$$f_{RF} = \left( INT + \frac{FRAC}{MOD} \right) \cdot \left( \frac{f_{pfd}}{DIV} \right) \tag{6}$$

Donde:

- $f_{RF}$  es la frecuencia de salida de RF.
- $INT$  es el factor de división entera.



- $FRAC$  es el numerador de la división fraccionaria. (0 a MOD-1)
- $MOD$  es el denominador de la división fraccionaria. (2 a 4095)
- $DIV$  es el divisor de la salida que divide la frecuencia del VCO. (1, 2, 4, 8, 16, 32 o 64)
- $f_{pfd}$  es la frecuencia de entrada del detector de fase.

Mientras que  $f_{pfd}$  se define como:

$$f_{pfd} = REF_{in} \cdot \frac{(1 + D)}{R(1 + T)} \quad (7)$$

Donde:

- $REF$  es la frecuencia de la señal de referencia (entrada al chip, al cual para este trabajo se emplea un oscilador de 10 MHz externo).
- $D$  es el duplicador de la frecuencia de entrada. (0 o 1)
- $R$  es el factor de división de referencia. (1 a 1023)
- $T$  es el factor de división por dos de la referencia. (0 o 1)

Es decir que, para lograr un barrido en frecuencia, es necesario generar un algoritmo que permita calcular de manera automática cada conjunto de registros para una frecuencia puntual, dentro del rango de trabajo. De esto se encarga el Firmware del PIC, quien recibe las frecuencias instantáneas del Software y empieza a calcular los registros para enviárselos a cada generador de RF.

#### 4.2.2. Esquemáticos

En el diagrama en bloques de la figura 16 se muestra la composición de la sección de Generadores de RF, el cual posee un oscilador de referencia de 10 MHz para ambos generadores de RF basados en el ADF4351, en conjunto con los buses de datos de cada chip para la carga de registros provenientes del PIC:

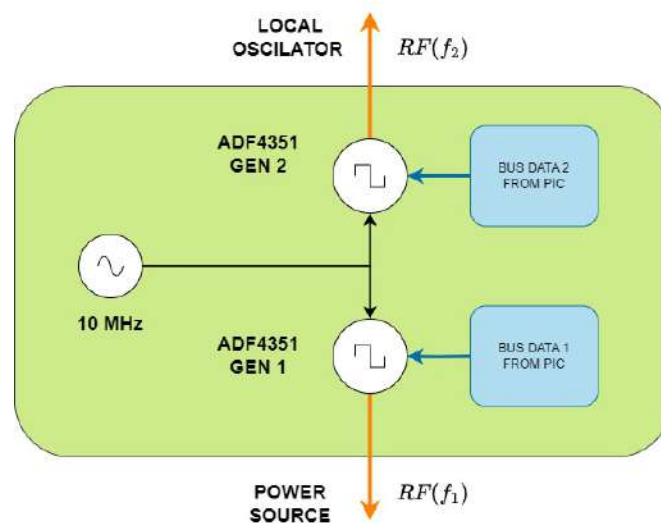


Figura 16: Esquema del doble generador ADF4351.

Para el oscilador de referencia de los chips se emplea un oscilador tipo Colpitts con cristal de 10 MHz y transistores TBJ NPN 2N3904 [17], dado que estos trabajan hasta 300 MHz. Además, posee un seguidor de tensión cuya única función es aumentar la ganancia de la señal.

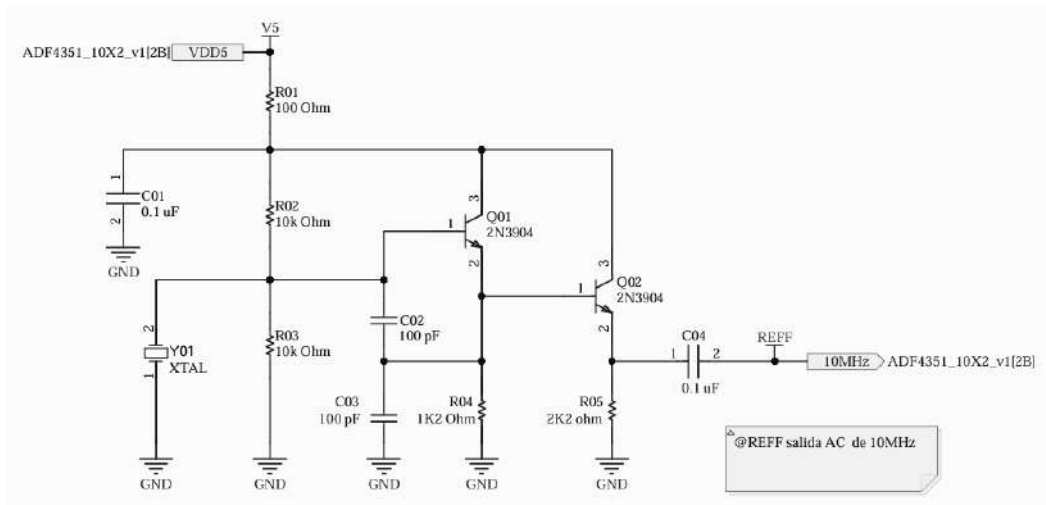


Figura 17: Oscilador Colpitts 10MHz basado en TBJ 2N3904.

La salida de este oscilador de referencia se conecta directamente a la entrada de cada uno de los ADF4351. Esto es posible gracias a que la entrada de referencia de estos chips es de alta impedancia, compatible con niveles TTL o CMOS, pero aquí se conectan directamente con desacople de continua. En la figura siguiente se observa la conexión típica recomendada por una hoja de aplicación del ADF4351 de propósito general, en el cual ya brinda el diseño del filtro del PLL y cuya salida, la cual es balanceada, se convierte a desbalanceada mediante un balún activo con una terminación a 50 Ω, implicando así dividir la potencia en dos, es decir, que la potencia disponible cae 3dB: de +5dBm a +2dBm. Ambos generadores son equivalentes.

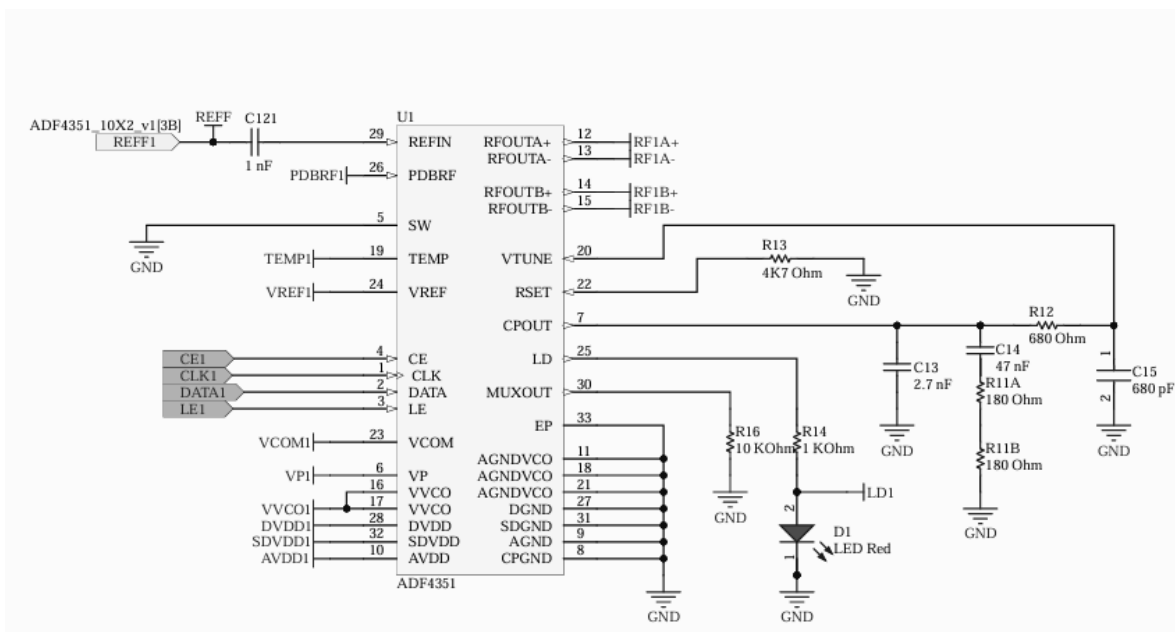


Figura 18: Generador 1 basado en chip ADF4351.

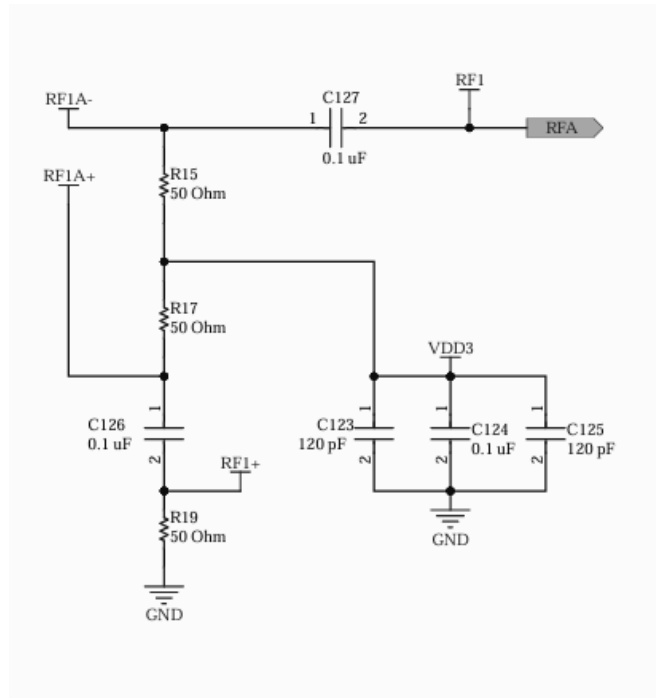


Figura 19: Balun para desbalancear la salida de RF.

Para suplimentar la corriente necesaria a las varias conexiones de tensión de cada chip ADF4351, la hoja de datos recomienda agregar diversos capacitores cerámicos lo más cerca posible del chip:

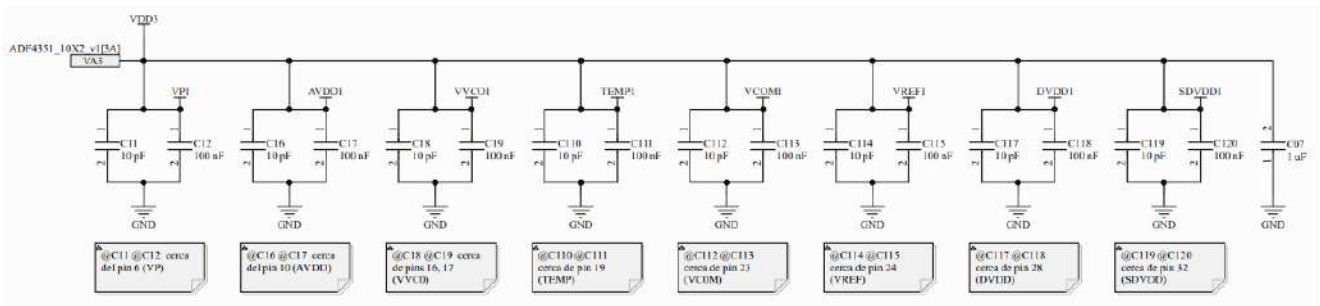


Figura 20: Decoupling capacitors cerca del chip ADF4351.

Por otro lado, cada bus de datos para la programación de los registros consta de una línea de sincronización de reloj CLK (de 20 MHz), una línea de datos DATA y dos líneas, CE (negada) y LE, las cuales son para la habilitación del chip (*chip enable*) y para la habilitación de carga de registros (*load enable*), respectivamente. Estas líneas son independientes entre cada chip, es decir, que hay 4 líneas por ADF4351.

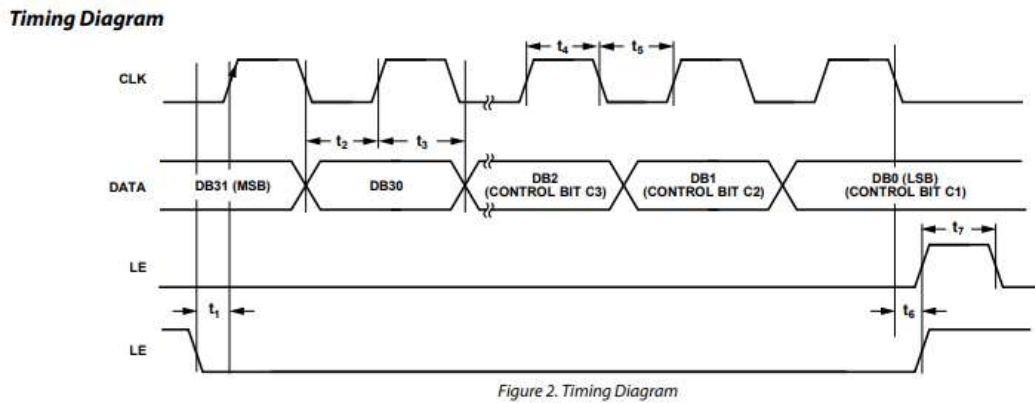


Figura 21: Diagrama temporal de las líneas digitales del chip ADF4351. Fuente: [16].

Las entradas de cada bus digital de los ADF4351 son de alta impedancia de tecnología CMOS, con niveles lógicos de 1,6 V (HIGH min) y 0,6 V (LOW max). Y dado que las salidas del PIC18F4550 aseguran niveles lógicos de 4,3 V (HIGH min) y 0,6 V (LOW max), es necesario agregar una red de atenuación unidireccional a cada línea:

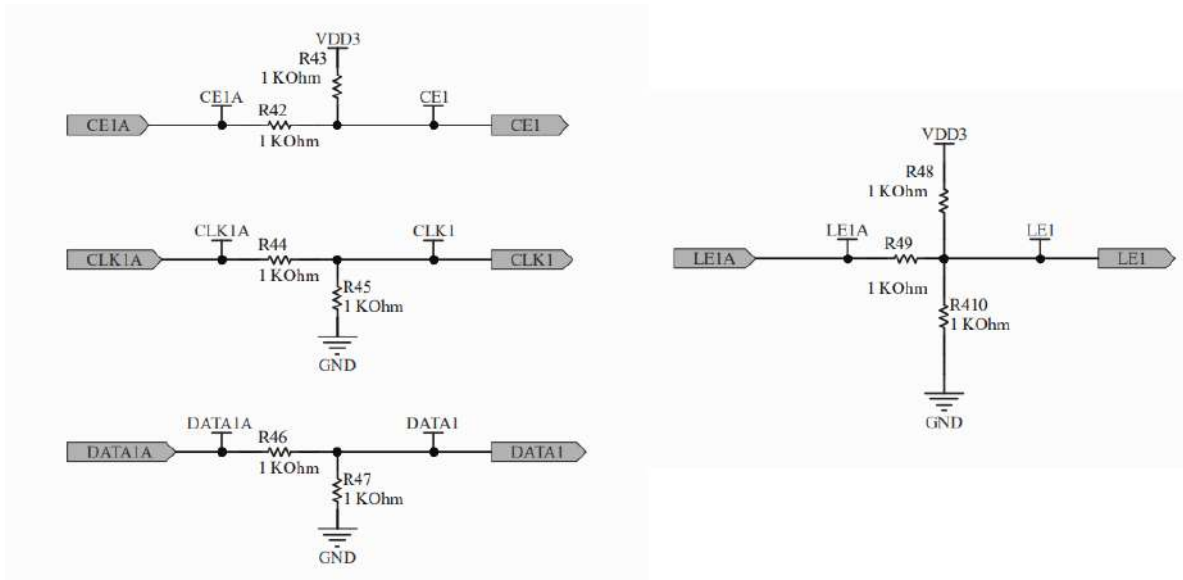


Figura 22: Conversor de niveles de lógica digital. Salidas PIC18F4550/ Entradas ADF4351.

En cuanto a la alimentación, hay que tener en cuenta que los chips ADF4351 necesitan una fuente estable de 3,3V, con un consumo aproximado de 100 mA cada uno. Y dado que se dispone de una fuente de 5V/2A para la alimentación de todo el circuito del VNA, se emplea el circuito integrado de regulación lineal UA78M (5V a 3,3V) con una capacidad máxima de 500mA, la cual es suficiente para el oscilador de referencia y los dos generadores ADF4351.

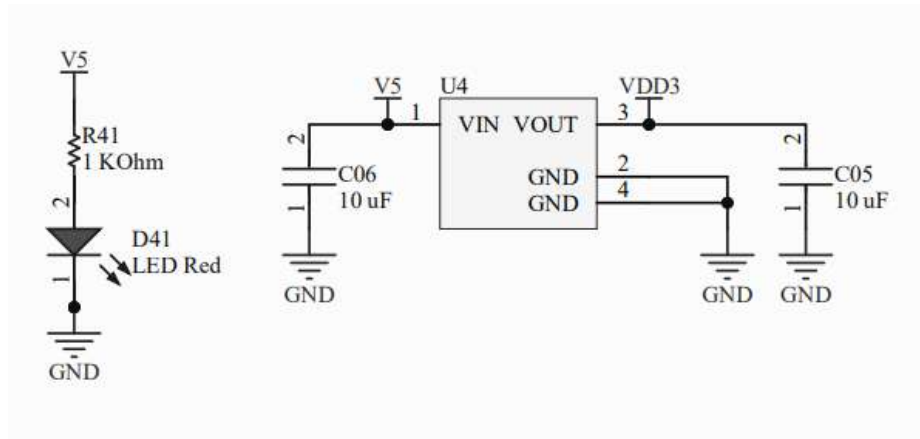


Figura 23: Regulador Lineal.

#### 4.2.3. PCB

En el Anexo 9 de este documento se encuentra información detallada acerca de los fundamentos y reglas que se emplean en el diseño de circuitos de RF. Para el diseño del PCB de este bloque se precisa un riguroso detalle en cuanto a la disposición de los componentes SMD y el empleo de tales reglas para la mitigación de efectos indeseados, como es la propia interferencia generada por los múltiples caminos de retorno, las inductancias parásitas, las autoresonancias, entre otros problemas. Para las líneas de transmisión se emplea la técnica de líneas coplanares [4], las cuales son menos susceptibles a interferencias que la técnica de Microstrip [4], en conjunto con las técnicas de *via stitching* y *via fencing* [18]. Para el cálculo del ancho del trazo se emplean las herramientas internas del programa Altium Designer [7].

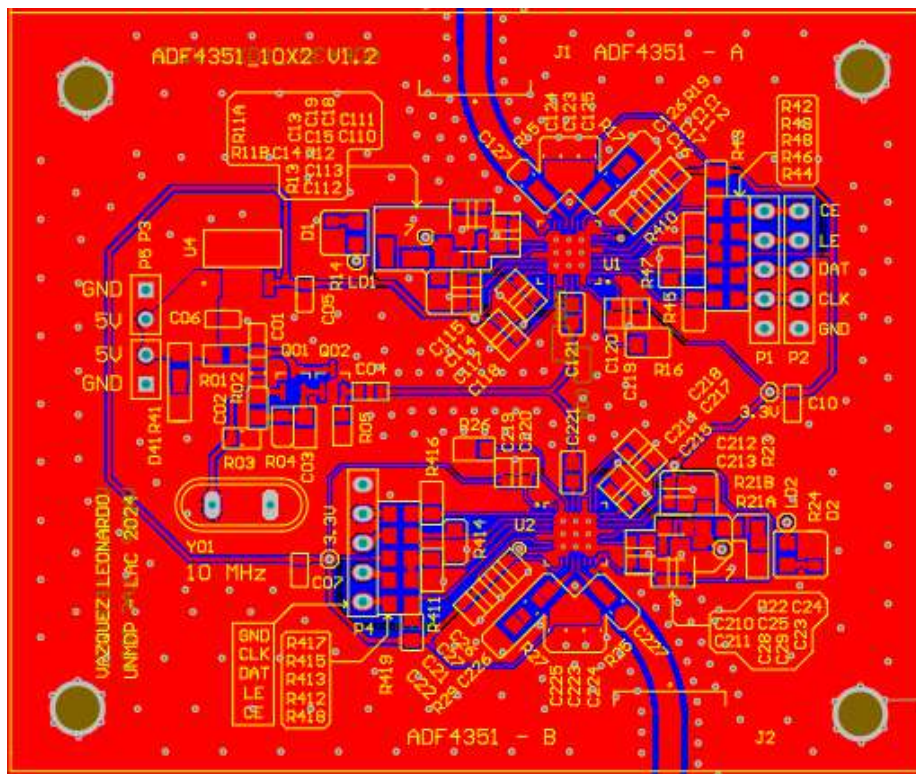


Figura 24: Diseño en PCB del bloque generadores.

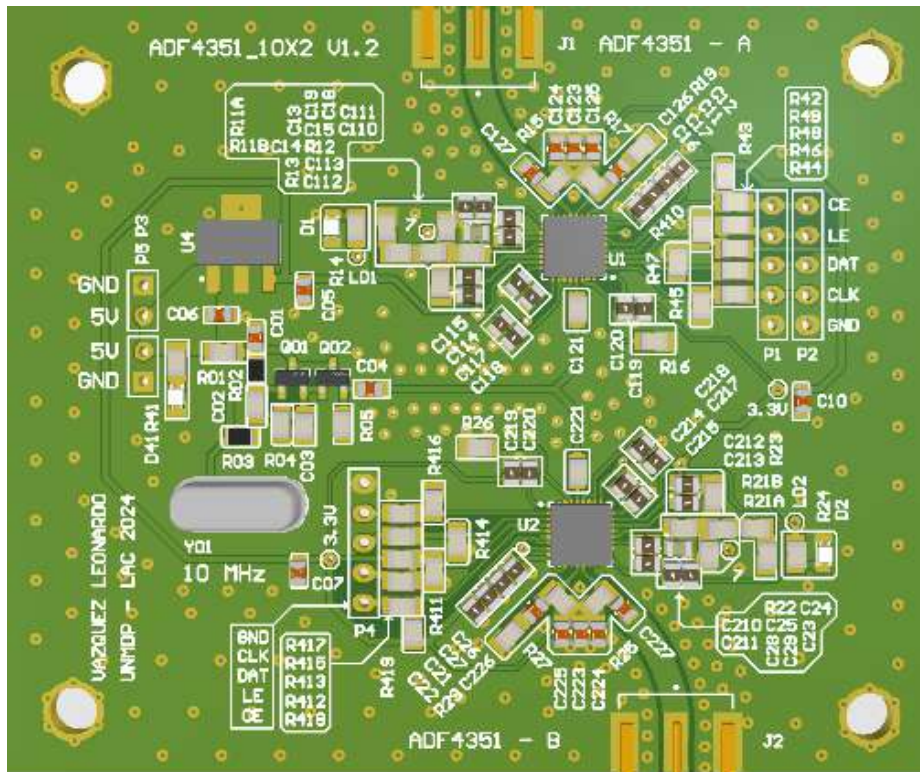


Figura 25: Diseño en PCB del bloque generadores.

### 4.3. Acoplador direccional resistivo

Mientras una de las señales del generador de RF se utiliza como oscilador local para la etapa de recepción, la otra señal restante se emplea para inyectar potencia al DUT. Por lo que, si se desea obtener información acerca de la señal que incide a este y la señal que se refleja al VNA, es necesario implementar un acoplador direccional.

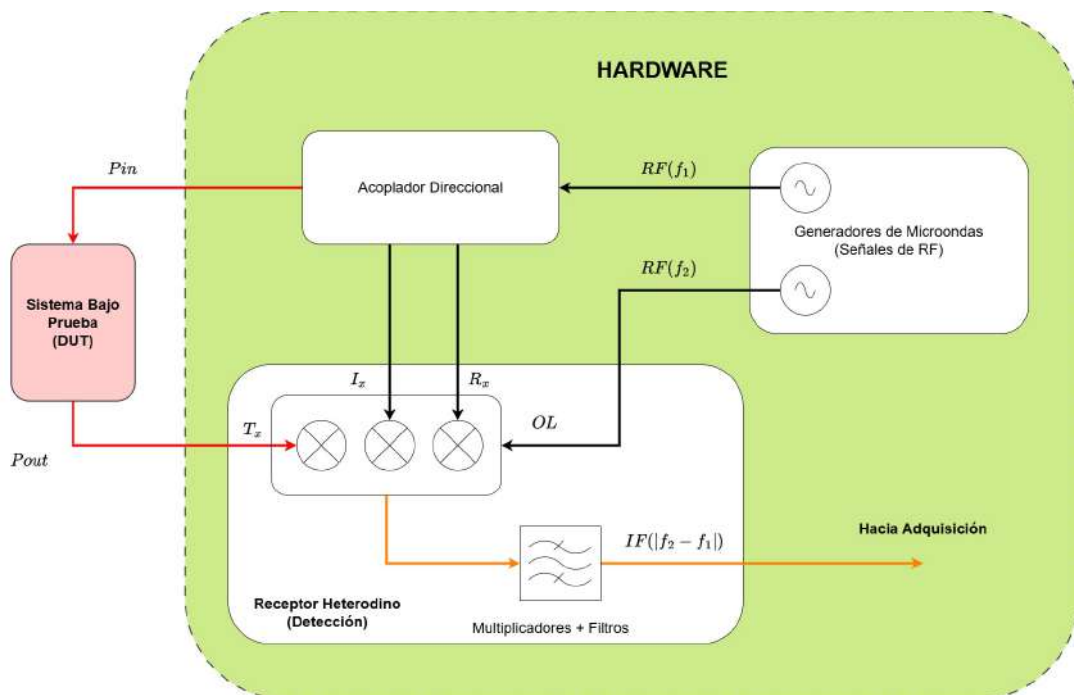


Figura 26: Etapa de RF y recepción.

Las especificaciones que debe cumplir el acoplador direccional son:

- Obtener una muestra de la señal incidente y una muestra de la señal reflejada en la banda de UHF.
- Obtener la mínima pérdida de inserción y la máxima directividad, si es posible.
- Todos los puertos deben trabajar con impedancia de  $50 \Omega$ .
- Se utilizarán como DUT a cargas corto circuito (CC), circuito abierto (CA) y  $50 \Omega$ .
- Adaptar las salidas de los puertos de acoplamiento a una etapa de banco de Mixers para su posterior procesamiento en baja frecuencia.

En la figura 27 se observa el diagrama del acoplador direccional propuesto en términos de parámetros concentrados.

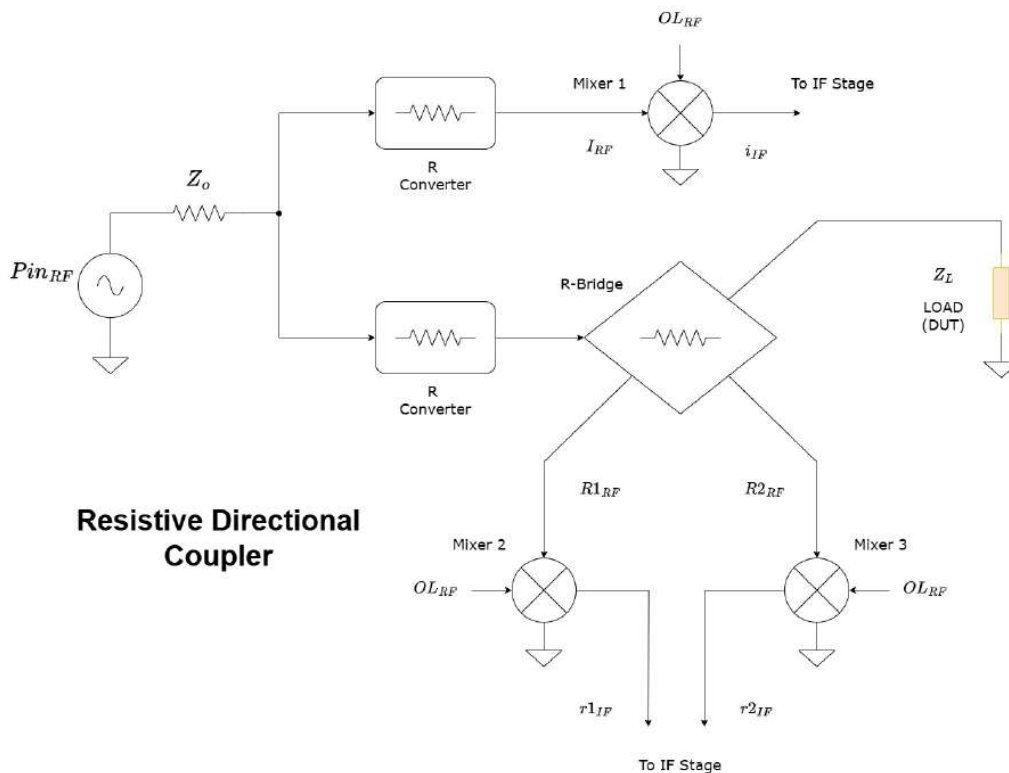


Figura 27: Acoplador direccional propuesto.

Teniendo como fuente de excitación a un generador de RF barredor de  $50 \Omega$ , se emplea un divisor de potencia resistivo para bifurcar el camino de la señal mediante dos ramas. La primera rama se emplea para tomar una muestra de la referencia o señal incidente hacia la próxima etapa mediante el primer Mixer. La segunda rama es la que se utiliza para el puente resistivo, que transfiere la potencia hacia el puerto de carga y entrega dos señales cuya diferencia representa a la señal reflejada, que van hacia la próxima etapa. En ambas ramas se emplean adaptadores de impedancia para minimizar las pérdidas de inserción.

En el Anexo 9 de este documento se encuentra en detalle el cálculo de cada uno de los componentes de este arreglo. Finalmente, el acoplador direccional, el cual está basado en un puente resistivo de Wheatstone [19], para su implementación en el analizador vectorial de redes es:

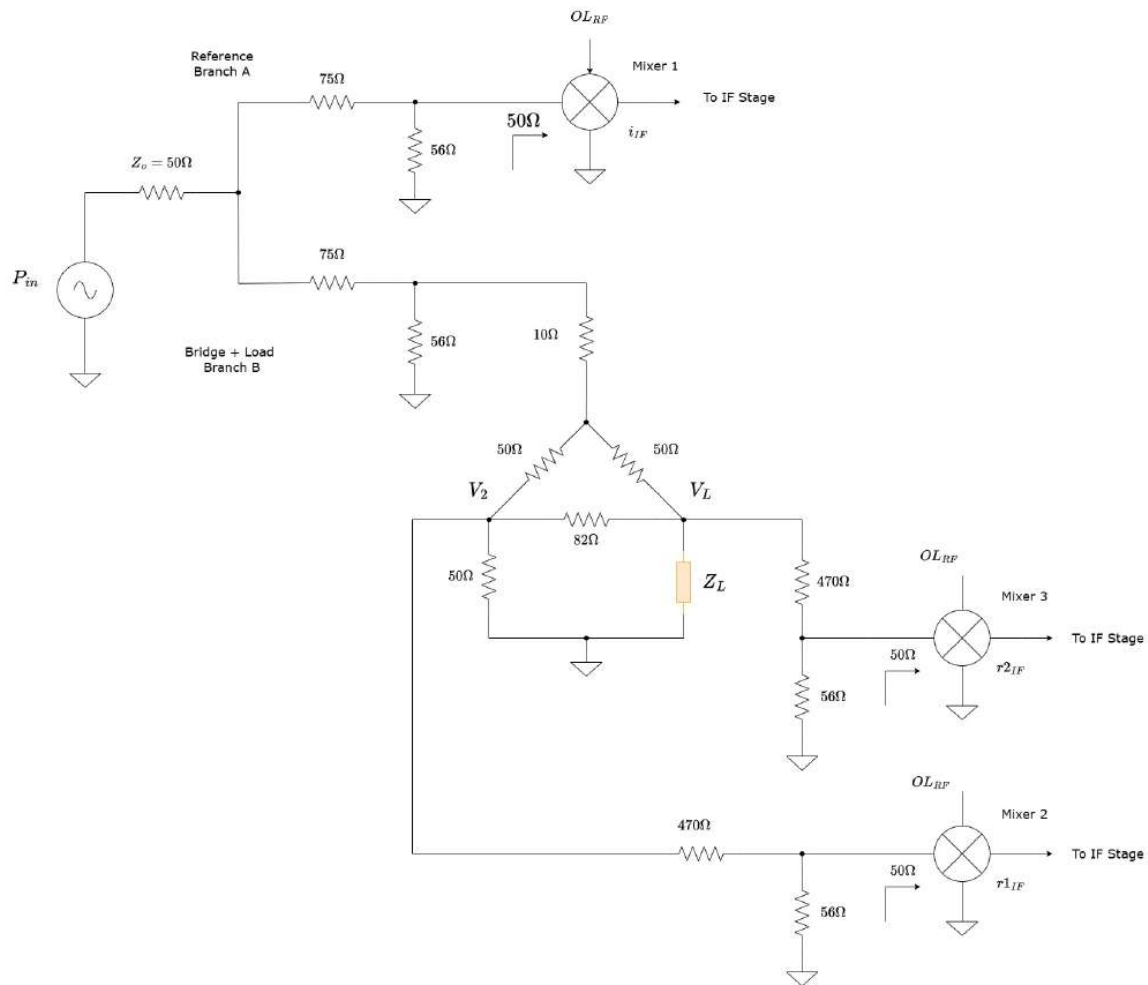


Figura 28: Acoplador direccional resistivo propuesto.

Desde el punto de vista de parámetros S, el acoplador propuesto queda con 5 puertos:

- Puerto 1: Puerto incidente.
- Puerto 2: Puerto pasante (hacia carga del DUT).
- Puerto 3: Puerto acoplado al puerto 1 (señal de referencia o muestra de incidente).
- Puerto 4a: Puerto acoplado al puerto 2 (medición de variación en carga referencia de 50  $\Omega$ ).
- Puerto 4b: Puerto acoplado al puerto 2 (medición de variación de carga del DUT).



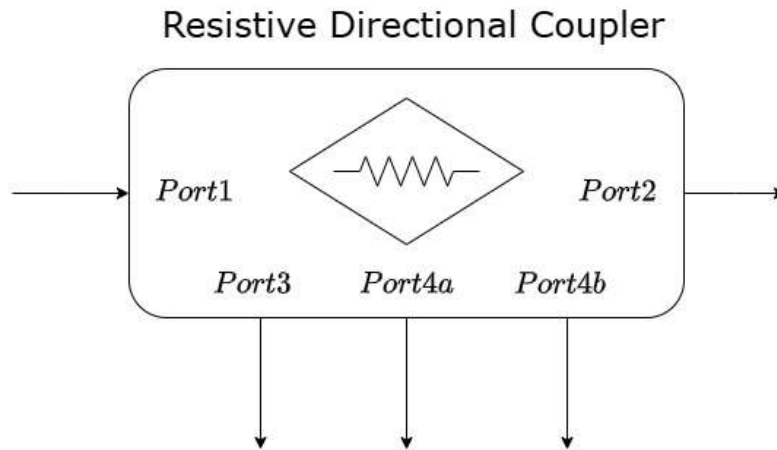


Figura 29: Diagrama en bloque del acoplador direccional resistivo propuesto.

Se puede demostrar en el anexo 9 que el coeficiente de reflexión del DUT, o simplemente el parámetro  $S_{11}$  o  $S_{22}$  del DUT, se obtiene a partir de los parámetros  $S_{4a}$  y  $S_{4b}$  del acoplador, que relacionan las tensiones de entrada a los Mixers con la señal de referencia o incidente:

$$\Gamma(DUT) = 286 \cdot (S_{4a} - S_{4b}) \quad (8)$$

A partir de los parámetros S (anexo 9) se pueden destacar algunas de las características más importantes de este acoplador direccional:

Pérdida de retorno (*Return loss*) Reflexión del generador de entrada:

$$R_L = -20 \cdot \log(|S_{11}(max)|) = 33dB \quad (9)$$

Pérdida de inserción (*Insertion loss*) Potencia inyectada sobre la carga:

$$I_L = -20 \cdot \log(|S_{51}|) = 19,2dB \quad (10)$$

Aislamiento (*Isolation*) Respecto del generador de entrada y la diferencia de los Mixers 2 y 3:

$$I = -20 \cdot \log(|S_{31} - S_{21}|) = 50dB \quad (11)$$

Acoplamiento (*Coupling*) Desde el generador de entrada hasta el Mixer 1 de referencia:

$$C = -20 \cdot \log(|S_{41}|) = 11,5dB \quad (12)$$

Directividad (*Directivity*): Relacionando la señal de referencia contra la señal proporcional a  $\Gamma$ :

$$D = 20 \cdot \log(|S_{31}/(S_{31} - S_{21})|) = 38,5dB \quad (13)$$

En la siguiente tabla se comparan algunos de los parámetros de las diferentes tecnologías de acopladores existentes con el acoplador propuesto:

Device	Directivity (dB)	Return Loss (dB)	Insertion Loss (dB)
Competitor Directional Coupler	19,3	14	0,2
Airline Coupler	23	17,5	0,1
BiDirectional Coupler	30,1	20,9	0,5
BiDirectional Bridge	37,3	19,3	1,9
Resistive Directional Coupler (this work)	38,5	33	19,2

Tabla 2: Comparación de acopladores. Fuente: [20]

A pesar de que los puentes poseen mayor pérdida de retorno e inserción que los acopladores convencionales, los primeros tienen mayor prestación en términos de directividad. Por otro lado, hay que tener en cuenta que el tamaño del acoplador también es una característica a considerar y también su respuesta en ancho de banda: mientras que la mayoría de los acopladores convencionales son de banda estrecha (alrededor del 10 % de la frecuencia central), los puentes resistivos, en teoría, son dispositivos de banda ancha.

#### 4.3.1. Esquemático

El esquemático del puente resistivo queda como:

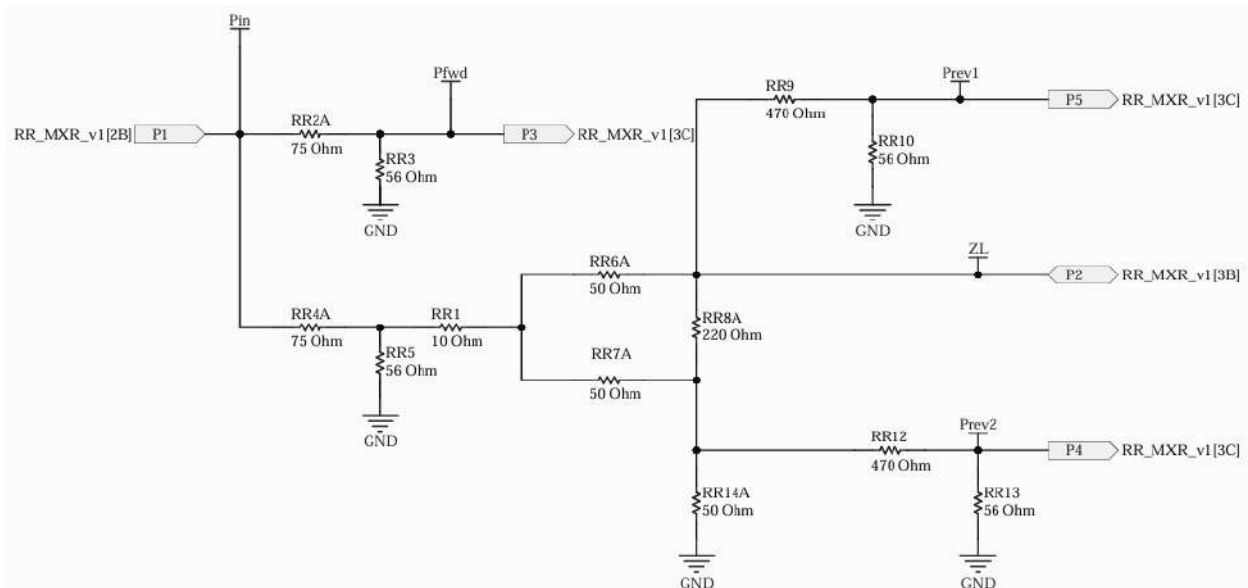


Figura 30: Esquemático del acoplador direccional resistivo.

#### 4.3.2. Diseño en PCB

Para el diseño en PCB hay que tener en consideración que debe adaptarse tanto a la entrada de potencia (desde el generador GEN1 de RF) como con la siguiente etapa de multiplicación (banco de Mixers). Esto implica que el posicionamiento de los componentes resistivos no es trivial, se diseña teniendo en cuenta algunas de las mejores prácticas en términos de diseño de RF y compatibilidad electromagnética: líneas de transmisión de 50  $\Omega$  coplanares con blindaje, caminos

de corrientes de retorno adecuados y planos de masa interconectados mediante varias vías. Los componentes resistivos son de tecnología SMD tamaño 0603. Se presupone idealidad en cuanto a su características en frecuencia.



Figura 31: *Layout* del acoplador direccional resistivo.

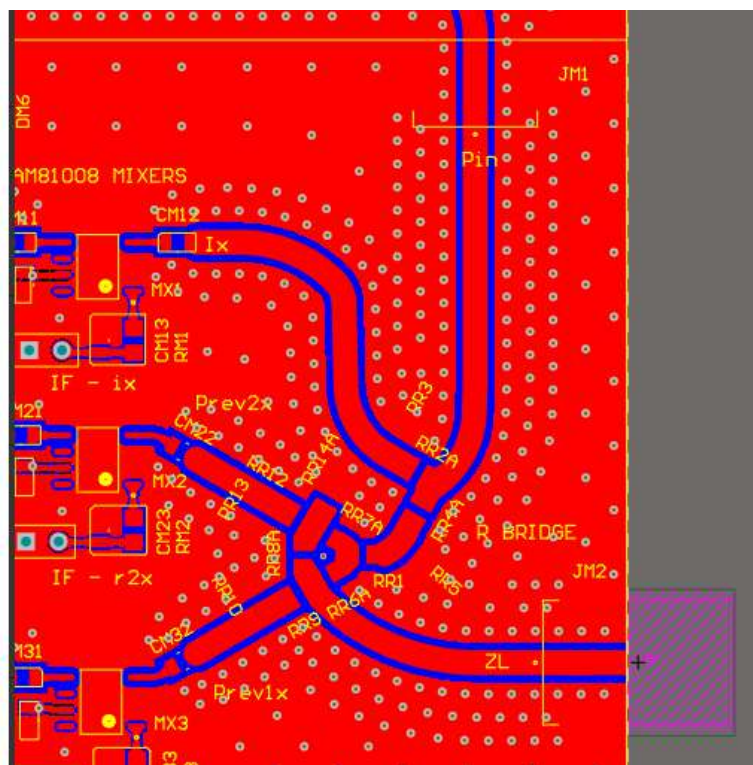


Figura 32: *Layout* del acoplador direccional resistivo en VNA.

En la figura 32 se puede observar, a la izquierda, la etapa de multiplicación del receptor; en centro, el arreglo resistivo; y a la derecha, el conector de carga ZL, o bien, el puerto de entrada 1 del DUT. Se emplean conectores SMA hembra de  $50 \Omega$  para superficies de PCB para la conexión entre el VNA y el DUT (puerto 1 de inyección de potencia y medición de reflexión).

#### 4.4. Receptor heterodino

En sistemas de comunicaciones se conoce como heterodinación [21] a aquella técnica en la cual se convierte una señal de alta frecuencia a otra de frecuencia intermedia o menor, mediante el uso de circuitos multiplicadores y un oscilador local. Se precisa entonces un receptor heterodino que sea capaz de convertir cuatro señales de RF (señales incidente, reflejadas y transmitida) a banda base, o bien frecuencias de audio, las cuales son mucho más sencillas de adquirir y procesar. Se propone entonces el siguiente banco, el cual emplea el generador 2 de frecuencia  $f_2$  como el oscilador local de un banco de cuatro Mixers, uno para cada señal de frecuencia  $f_1$ . Luego, en la etapa de IF, se utiliza un filtro pasa-bajos para utilizar únicamente a la componente de  $f_1 - f_2$  (frecuencias de audio) y no a la de  $f_1 + f_2$  (alta frecuencia).

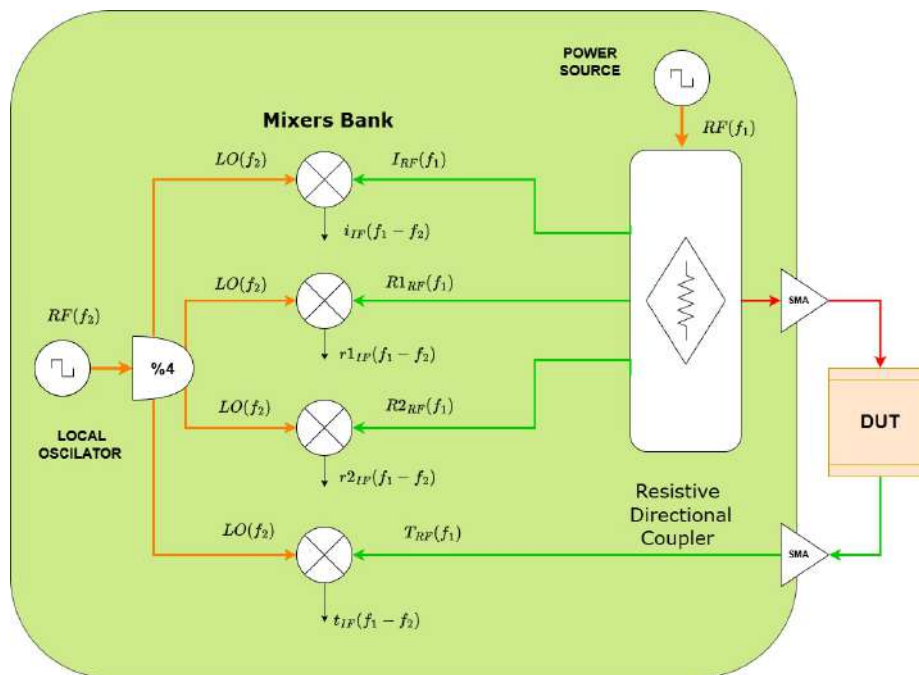


Figura 33: Receptor heterodino propuesto.

##### 4.4.1. Multiplicadores

Para el banco de multiplicadores se emplean los circuitos integrados IAM8001 [22], los cuales son Mixers basados en chips MMIC con excelentes prestaciones de ancho de banda, que van desde 50 MHz a 5 GHz, y con la característica de que trabajan con una impedancia característica de entrada y salida de  $50 \Omega$ , lo cual facilita el diseño por etapas de un sistema de comunicaciones.

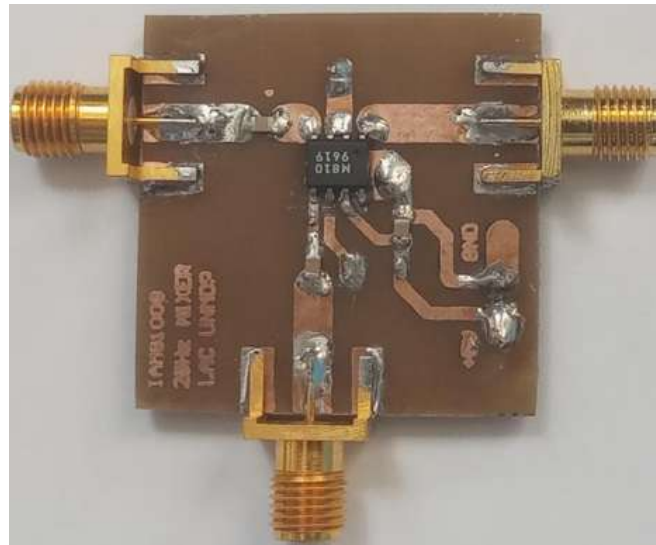


Figura 34: Mixer IAM81008. Prototipo.

#### 4.4.2. Divisor de potencia

Para poder dividir la potencia de la señal del oscilador local y repartirla de manera equitativa hacia cada uno de los mixers, se emplea la siguiente topología [23]:

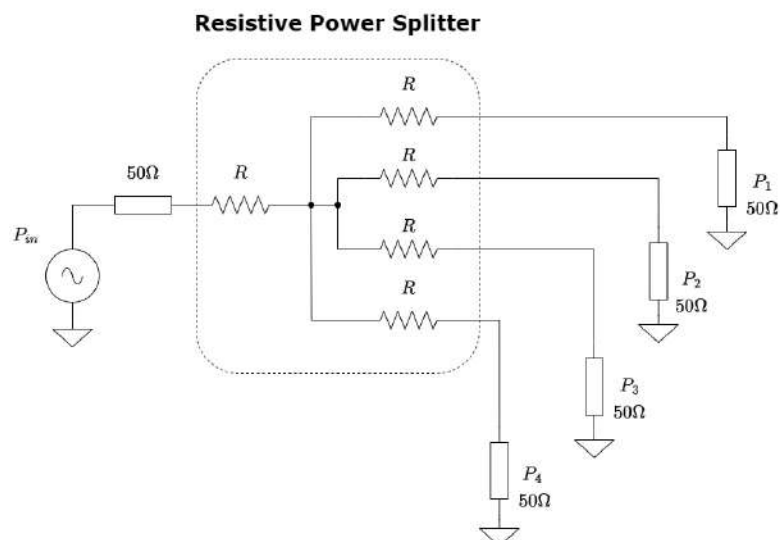


Figura 35: Divisor de potencia resistivo.

La impedancia vista por cada puerto debe estar adaptada a  $50 \Omega$  para maximizar la transferencia de potencia, y dado que la impedancia de entrada de los mixers es efectivamente de  $50 \Omega$ , al igual que la salida del generador ( $Z_o$ ), facilita así el diseño del divisor. Puede demostrarse que, para un divisor de 5 ramas ( $N = 4$ ) y adaptación a  $Z_o = 50 \Omega$ , el valor del resistor es:

$$R = Z_o \cdot \frac{(N - 1)}{(N + 1)} = 30\Omega \quad (14)$$

En la cual, mientras la mitad de la potencia se disipa en los resistores, la otra mitad de la potencia se subdivide en cada rama (atenuación de 12 dB), y dado que hay adaptación en cada rama, si el generador nos entrega +2 dBm, en cada mixer se inyecta una potencia de oscilador local:

$$P_{in} = +2dBm - 12dB = -10dBm \tag{15}$$

El Mixer IAM81008 posee una ganancia de conversión (RF-IF) que depende de la señal de oscilador local:

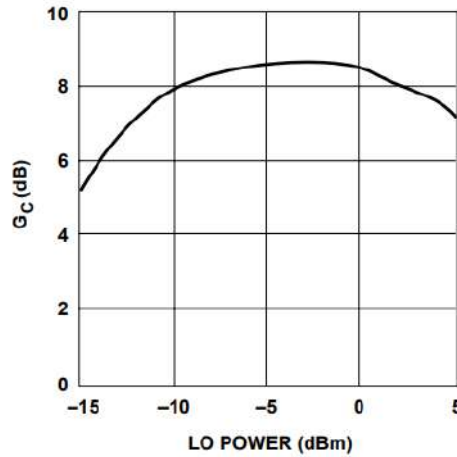


Figure 5. RF to IF Conversion Gain vs. LO Power.

Figura 36: Ganancia Mixer IAM81008. Fuente: [22].

Por lo que la ganancia de conversión es  $G = 8dB$ .

#### 4.4.3. Esquemáticos

En la figura 37 se observa el esquemático de uno de los cuatro mixers, el cual a la salida de cada uno hay un filtro pasabajos RC de 16 kHz. Esto es debido a que las frecuencias de audio se definen aproximadamente desde  $DC$  hasta 20 kHz. De esta manera, la señal resultante es la componente de la diferencia  $f_1 - f_2$ . Los capacitores restantes que se agregan son para desacople de continua.

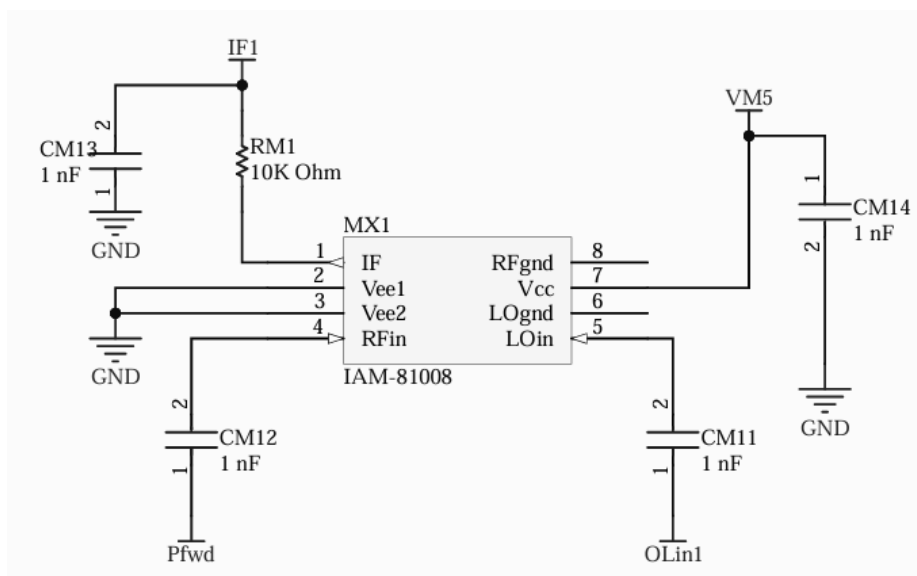


Figura 37: Esquemático Mixer IAM81008.

Para el divisor de potencia resistivo se emplean resistores comerciales SMD de  $27\ \Omega$  (a fines prácticos no influye mucho la diferencia entre  $27\ \Omega$  y  $30\ \Omega$ )

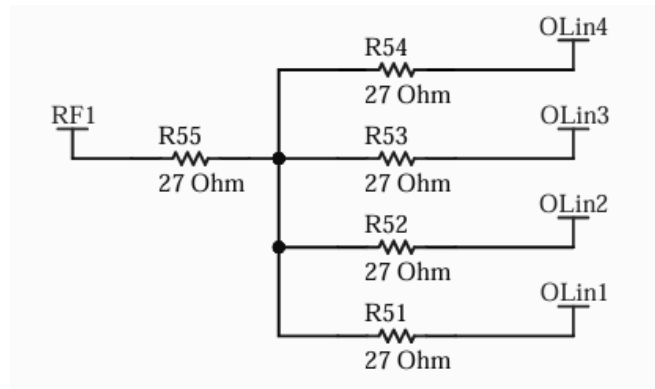


Figura 38: Esquemático divisor de potencia resistivo.

#### 4.4.4. Diseño de PCB

Se muestra el diseño en PCB del conjunto divisor de potencia resistivo-banco de mixers-acoplador direccional resistivo-conectores SMA (DUT), es decir, del receptor heterodino junto al acoplador direccional resistivo.

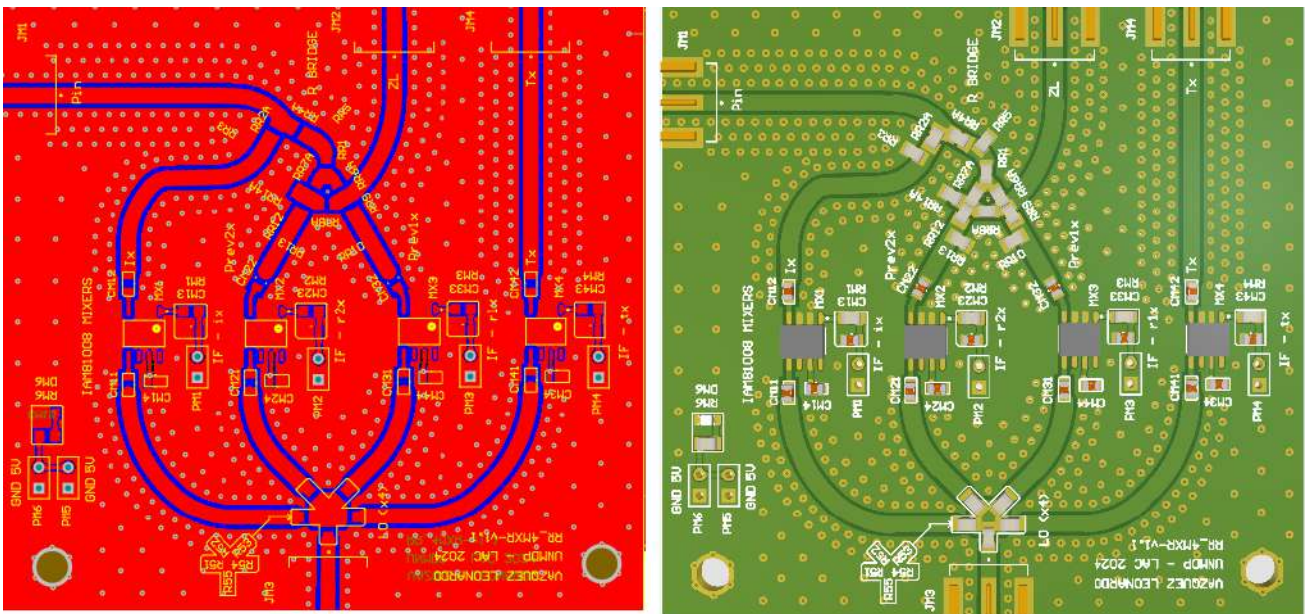


Figura 39: Diseño en PCB del receptor heterodino junto al acoplador.

Para conectar las señales de frecuencia intermedia IF hacia la siguiente etapa se colocan simplemente postes macho para cables tipo *jumper*. Se emplean dos debido a una posible fuga de RF (un cable es de retorno) que se desea evitar en la etapa de adquisición.

## 4.5. Adquisición

A partir de las señales de IF del receptor heterodino, el sistema debe ser capaz de adquirir y digitalizar dichas señales para poder procesarlas y así calcularles la amplitud y fase, para luego obtener así el parámetro S correspondiente. Para ello se emplean dos adquirentes de audio o simplemente CODECs:

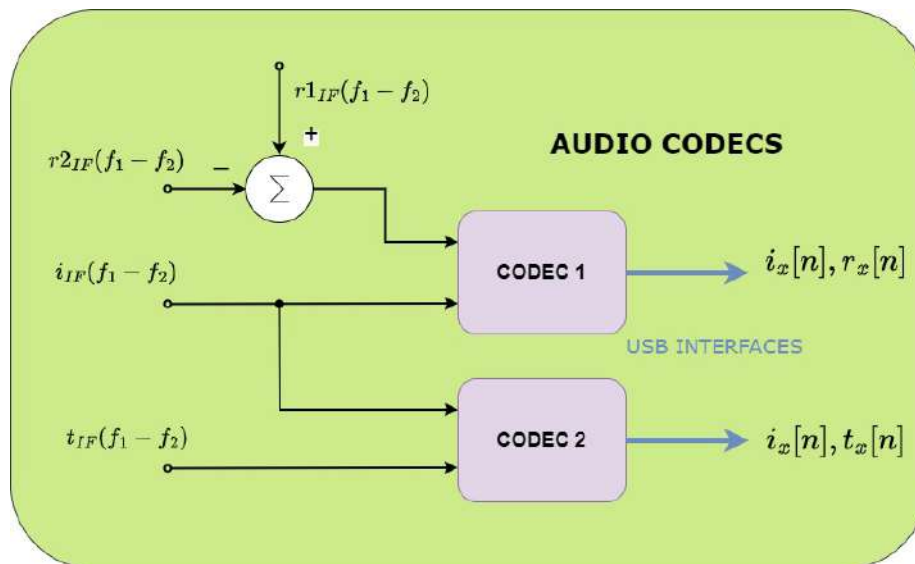


Figura 40: Diagrama en bloques de la etapa de adquisición.

La razón por la cual se emplean dos adquirentes tiene que ver con el uso de los chips PCM2904 [24], los cuales son unos CODECs de audio económicos con interfaz USB incluida, pero que poseen 2 canales de medición. Y dado que al menos se precisan 3 mediciones (incidente, reflejada y transmitida), se utilizan dos codecs al cual uno llamaremos simplemente como de reflexión (S11 o S22) y al otro de transmisión (S12 o S21). Para realizar la diferencia entre las señales provenientes del acoplador se emplea un restador, al cual se sintetiza mediante un amplificador operacional.

### 4.5.1. CODECs de audio

Entre las características principales de los chips PCM2904 se encuentran:

- Comunicación con protocolo USB 1.1 (+5V)
- Conversores ADC Sigma-Delta de 16 bits. 2 canales (*left and right*).
- Tasa de muestreo: 48 kHz
- Rango dinámico de entrada:  $\pm 1V_{rms} = \pm 2,82V_{pp}$
- Impedancia de entrada: 30 k $\Omega$



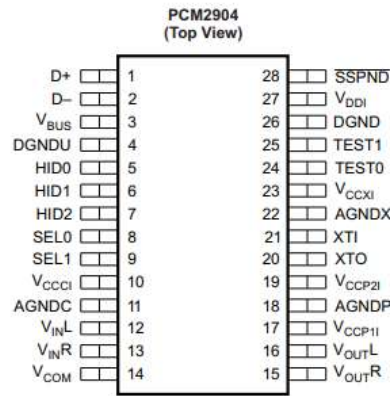


Figura 41: Pines del chip PCM2904. Fuente: [24].

En la entrada del ADC (figura 42), se encuentra una etapa de alta impedancia seguida de un filtro pasabajos, dándole una característica ideal para la adquisición de señales dentro del rango de frecuencias de audio, que cubre desde *DC* hasta 20 kHz, aproximadamente (figura 43).

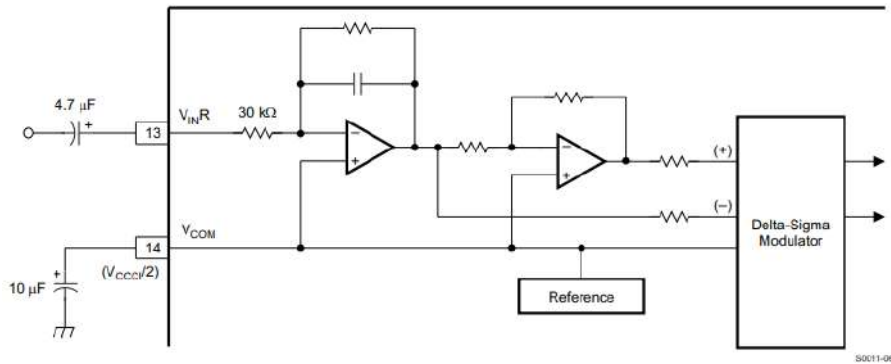


Figura 42: Entrada de cada ADC del PCM2904. Fuente: [24].

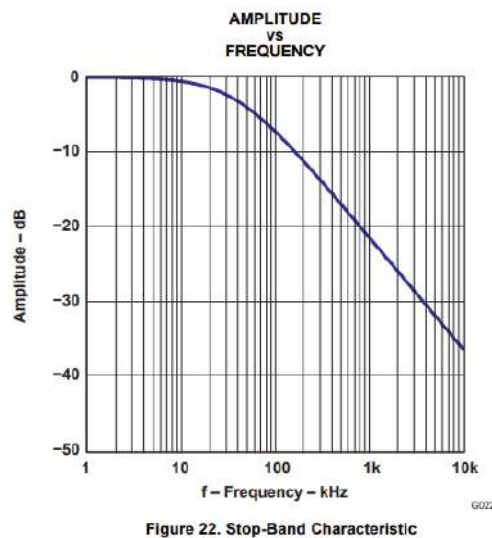


Figura 43: Característica de filtrado de cada ADC del PCM2904. Fuente: [24].

### 4.5.2. Restador

Para implementar el restador se emplea un circuito integrado MCP6023 [25] el cual es un amplificador operacional *rail-to-rail* para emplearlo entre 0V y 5V. Entre sus características más importantes se destaca: es un circuito integrado de bajo ruido y bajo consumo, con ancho de banda de 10 MHz, alta impedancia en sus entradas y una ganancia a lazo abierto de 135dB, suficiente para realizar una ganancia de lazo cerrado más chica mediante realimentación negativa.

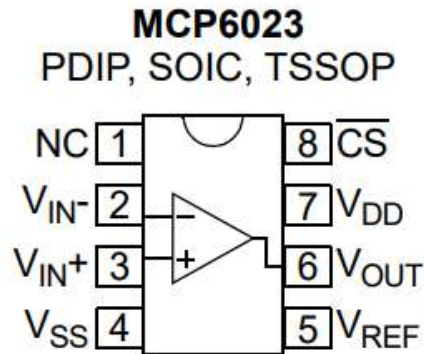


Figura 44: Pines del chip MCP6023. Fuente: [25].

### 4.5.3. Recepción y adquisición

Ahora bien, hay que corroborar que los niveles de tensión de las 4 señales en cuestión se encuentren dentro del rango dinámico de la etapa de adquisición. Por otro lado, es de esperar incluso que al realizar la resta entre  $r_1$  y  $r_2$ , dicha diferencia sea pequeña por lo que sería prudente agregar también una etapa de amplificación en el restador. La figura 45 muestra la etapa intermedia entre el receptor heterodino y la etapa de adquisición:

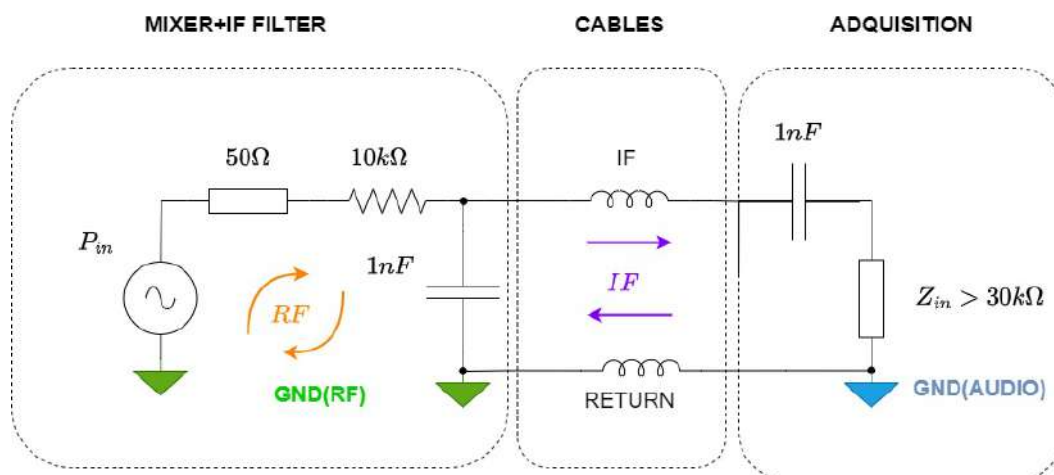


Figura 45: Etapa de recepción y adquisición.

El propósito de los cables es claramente generar un camino de ida y vuelta (todas las corrientes en loop) para las componentes de frecuencias intermedias  $IF$ , que van desde la salida del Mixer hasta la etapa de recepción (restador y/o CODECs). Cualquier componente de alta frecuencia verá un camino de retorno debido a la alta impedancia del filtro, por lo que se reflejará completamente. Esto también podría ocurrir debido a la característica inductiva del cable empleado.

Para cada señal que sale de cada Mixer, habrá una transferencia que depende de la carga en el puerto  $Z_L$  (conexión VNA DUT entrada):

$$P_s(\text{dBm}) = +2\text{dBm}(\text{GEN1}) - S_x(\text{acoplador}) + 8\text{dB}(\text{Gain}) \quad (16)$$

Para convertir la potencia en dBm en tensión se emplea la siguiente relación :

$$V_{pp}(50\Omega) = \sqrt{8 \cdot 50\Omega \cdot 10^{-3}\text{W} \cdot 10^{P_s/10}} \quad (17)$$

No obstante, este valor corresponde a un sistema de  $50 \Omega$ . La potencia o tensión disponible en realidad es el doble (HIGH Z). En la siguiente tabla se resumen los posibles valores en potencia y tensiones para cada una de las 4 señales. Para el caso de la señal transmitida, se interconecta mediante un cable ideal THR (*Through*) al puerto  $Z_L$  (entrada al DUT) y al puerto  $T_x$  (salida del DUT).

Signal (IF) Output	CC			50 Ohm			CA		
	dBm	mVpp (50 Ohm)	mVpp (High Z)	dBm	mVpp (50 Ohm)	mVpp (High Z)	dBm	mVpp (50 Ohm)	mVpp (High Z)
ix	-1.7	520	1040	-1.6	526	1052	-1.5	532	1064
r1	-39	7	14	-35	11	22	-32	16	32
r2	-inf	0	0	-35	11	22	-30	20	40
r1 - r2	-39	7	14	-inf	0	0	-16	4	8
tx (thr)	-	-	-	-9	225	450	-	-	-

Tabla 3: Valores de las señales a adquirir.

Se observa que todos los valores están dentro del rango de adquisición de los PCM2904. No obstante, la diferencia  $r_1 - r_2$  es pequeña por lo que, para obtener una mayor resolución en la adquisición se agrega una ganancia de 10 veces en el restador implementado con el amplificador operacional, al cual se le agrega entonces una ganancia mediante un lazo de realimentación con componentes resistivos y un offset para que su salida excursione en valores positivos.

#### 4.5.4. Esquemáticos

Se muestran los esquemáticos correspondientes al restador basado en el MCP6023 y a uno de los codecs PCM2904, los cuales son idénticos. Para la interfaz USB se agregan conectores USB 2.0 hembra tipo B.

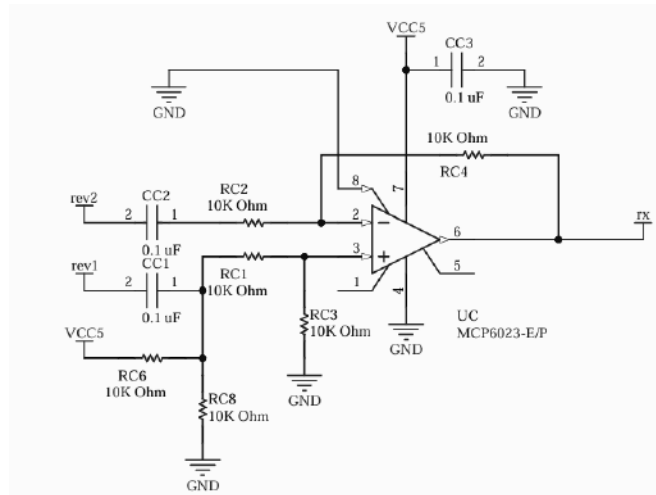


Figura 46: Esquemático del circuito restador.

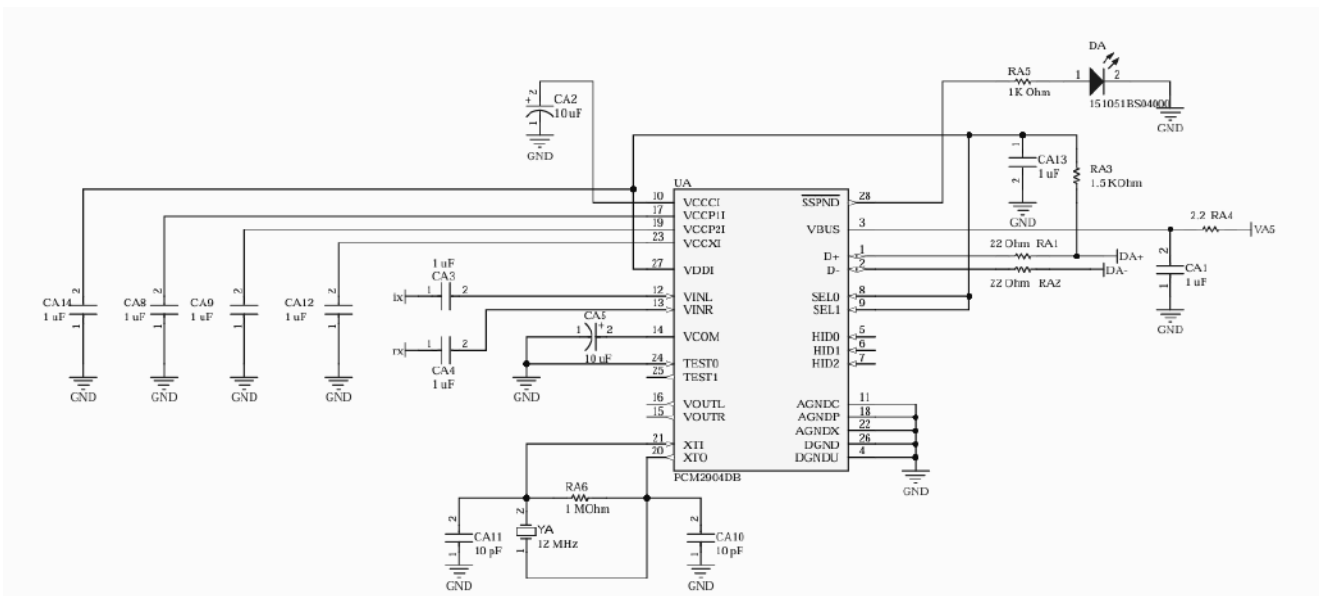


Figura 47: Esquemático de uno de los adquirentes CODEC.

#### 4.5.5. Diseño en PCB

Para el diseño en PCB simplemente se colocan de manera simétrica ambos adquirentes, de los cuales uno recibirá las señales de los parámetros de reflexión (S11 o S22) y otro las señales de transmisión (S21 o S12).

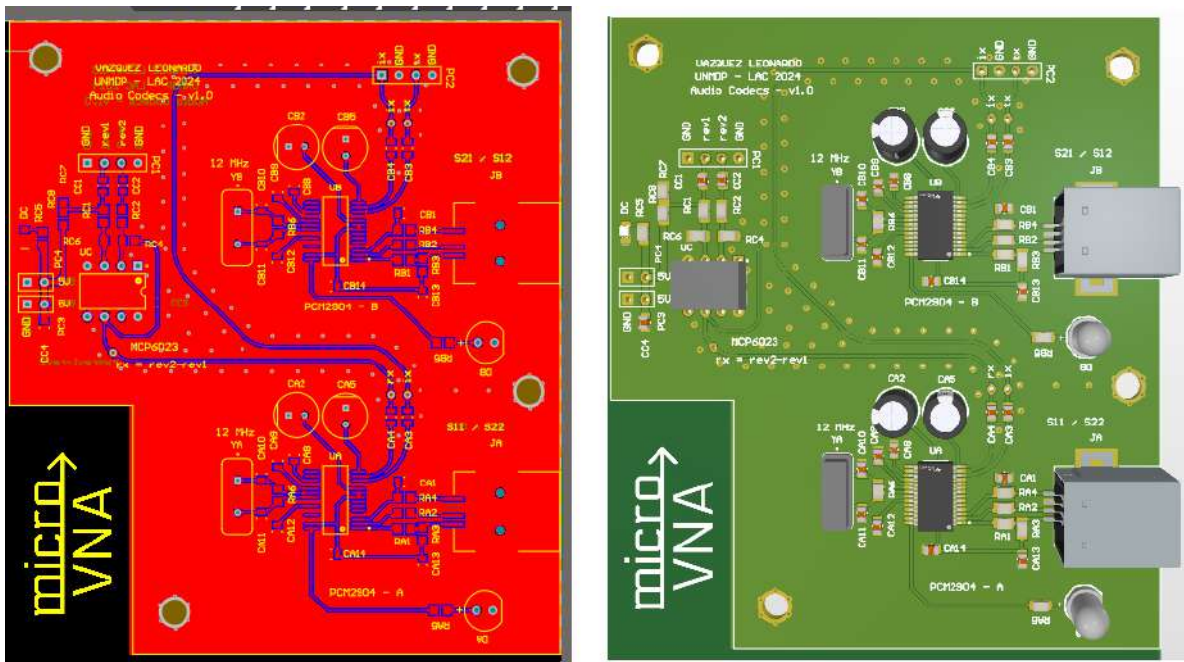


Figura 48: Diseño en PCB de la etapa de adquisición.

## 4.6. Ecuaciones Finales

### 4.6.1. Coeficientes de reflexión S11 y S22

El coeficiente de reflexión del DUT medido por el acoplador direccional 9 resistivo se calcula como:

$$\Gamma(DUT) = 286 \cdot (S_{4a} - S_{4b}) \quad (18)$$

Pero los parámetros S mencionados son respecto a la transferencia desde la entrada del acoplador hasta la entrada de los Mixers:

$$S_{4a} = \frac{V_{in}(MIX2)}{V_{in}} \quad (19)$$

$$S_{4b} = \frac{V_{in}(MIX1)}{V_{in}} \quad (20)$$

No obstante, la señal  $V_{in}$  se mide con el primer Mixer pero está  $-12\text{dB}$  por debajo, es decir que la entrada del Mixer 1 es:

$$V_i(RF) = \frac{V_{in}}{4} \quad (21)$$

Las señales de frecuencia intermedia están afectadas por la ganancia de conversión de los Mixers  $+8\text{dB}$ . Por otro lado, el restador también posee una ganancia de 10 veces, lo cual, desde la etapa de adquisición el cociente entre las mediciones del CODEC es:

$$\frac{\text{codec}(r_1 - r_2)}{\text{codec}(i_x)} = \frac{10 \cdot (V_{r1} - V_{r2})(IF)}{V_i(IF)} = \frac{10 \cdot (V_{r1} - V_{r2})(RF) \cdot (G_{MIX})}{V_i(RF) \cdot (G_{MIX})} \quad (22)$$

pero  $V_{in} = 4 \cdot V_i(RF)$ , lo cual, la medición de  $\Gamma$  que se realiza en la etapa de procesamiento es:

$$\frac{\text{codec}(r_1 - r_2)}{\text{codec}(i_x)} = 40 \cdot \frac{(V_{r1} - V_{r2})(RF)}{V_{in}(RF)} = 40 \cdot (S_{4a} - S_{4b}) \quad (23)$$

De esta manera, a partir de las mediciones del CODEC de reflexión se relaciona el coeficiente  $\Gamma$  del DUT como:

$$|\Gamma(DUT)| = \frac{286}{40} \cdot \frac{|\text{codec}(r_1 - r_2)|}{|\text{codec}(i_x)|} \quad (24)$$

$$\arg(\Gamma)(DUT) = \arg(\text{codec}(r_1 - r_2)) - \arg(\text{codec}(i_x)) \quad (25)$$

#### 4.6.2. Coeficientes de transmisión S21 y S12

Realizando un análisis similar, se define el coeficiente de transmisión del DUT como la transferencia en tensión de la señal incidente y transmitida a este último:

$$G(DUT) = \frac{Vt_x(RF)}{Vi_x(RF)} \quad (26)$$

No obstante, la señal que se inyecta al DUT es la señal que viene desde el generador 1 y pasa por el acoplador direccional resistivo, el cual posee una pérdida de inserción de +20dB aproximadamente, o lo que es lo mismo:  $Vi_x(RF) = V_{in}/100$ . De esta manera, las mediciones del CODEC de transmisión se definen como:

$$\frac{\text{codec}(t_x)}{\text{codec}(i_x)} = \frac{Vt_x(IF)}{Vi(IF)} = \frac{Vt_x(RF) \cdot (G_{MIX})}{Vi(RF) \cdot (G_{MIX})} \quad (27)$$

pero  $V_{in} = 4 \cdot Vi(RF)$  y  $Vi_x(RF) = V_{in}/100$ , lo cual:

$$\frac{\text{codec}(t_x)}{\text{codec}(i_x)} = \frac{Vt_x(RF)}{Vi_x(RF) \cdot 100/4} \quad (28)$$

Finalmente, a partir de las mediciones del CODEC de transmisión se relaciona la transferencia  $G$  del DUT como:

$$|G(DUT)| = \frac{100}{4} \cdot \frac{|\text{codec}(t_x)|}{|\text{codec}(i_x)|} \quad (29)$$

$$\arg(G)(DUT) = \arg(\text{codec}(t_x)) - \arg(\text{codec}(i_x)) \quad (30)$$

## 4.7. Placa completa

### 4.7.1. Diseño en PCB

En la figura siguiente se observa el diseño completo de la placa con todas las etapas. El tamaño de la misma es de 18cm x 18cm. El material utilizado es el ya conocido FR4, con espesor de cobre de  $35\mu\text{m}$  y máscara antisoldante. Para las conexiones de RF se emplean líneas de transmisión coplanares. Se agregaron perforaciones para mitigar efectos indeseados de RF. Además, los planos de masa de las etapas microcontrolador, RF y adquisición están conectadas en puntos puntuales para evitar propagaciones indeseadas de RF.

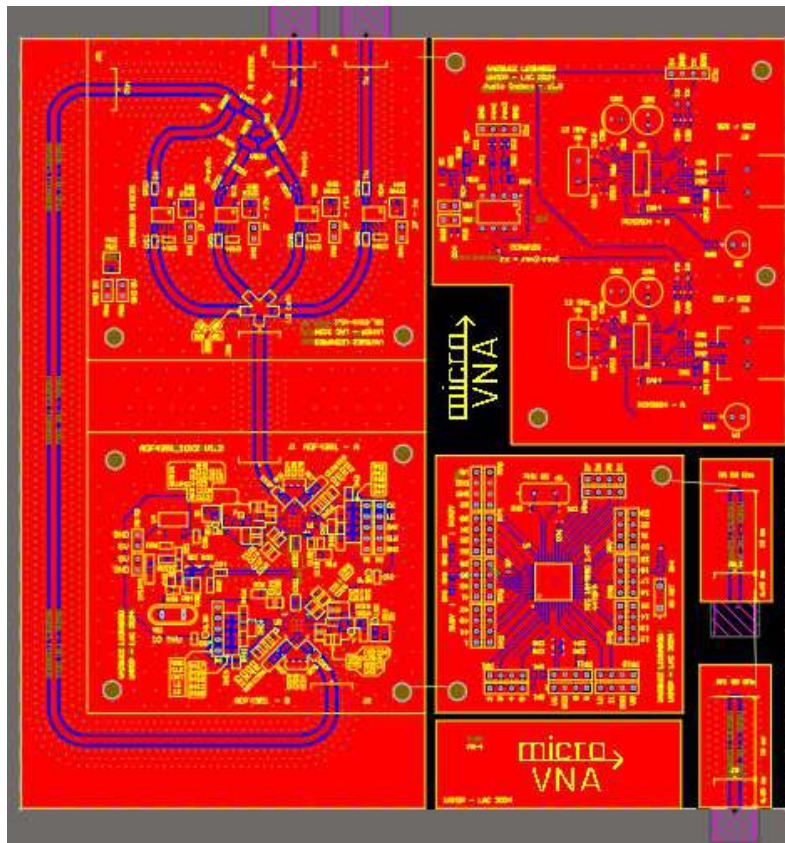


Figura 49: Diseño de *layout* en PCB del VNA.

### 4.7.2. Placa Final

En la figura siguiente se observa la placa final sin los componentes soldados. La misma fue diseñada por el tesista y construida por un fabricante de placas PCB de la industria nacional:

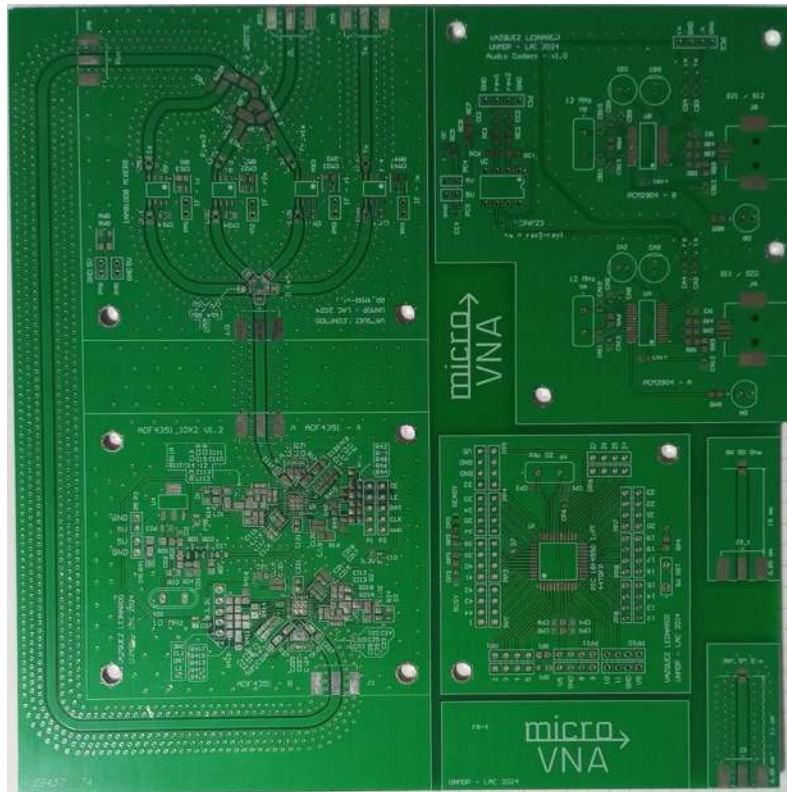


Figura 50: Placa PCB del VNA. Sin componentes.

Para colocar cada uno de ellos se debe asegurar que la temperatura del soldador y el estaño no supere los máximos de temperatura de cada chip y cada dispositivo.

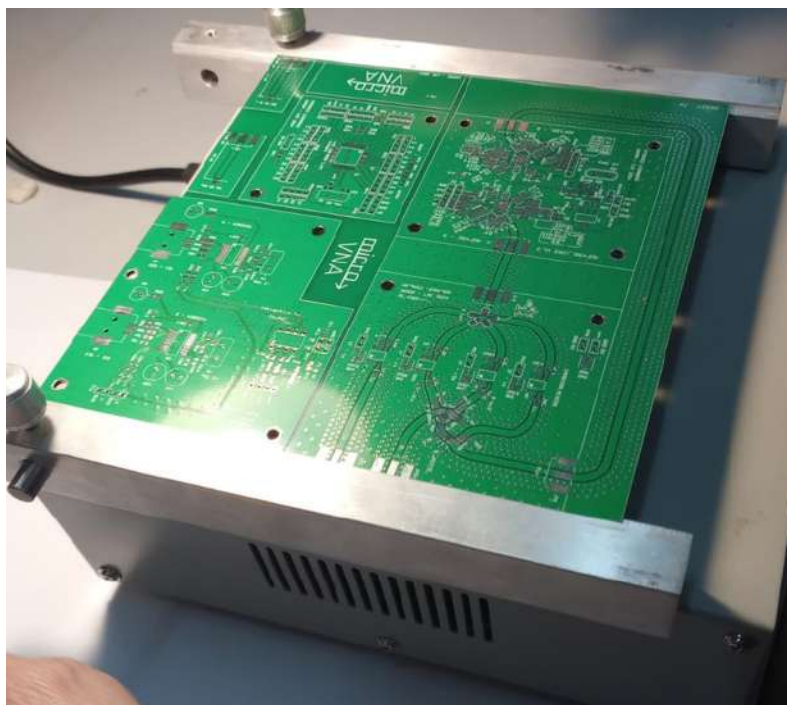


Figura 51: Placa PCB del VNA en máquina precalentadora para soldar.



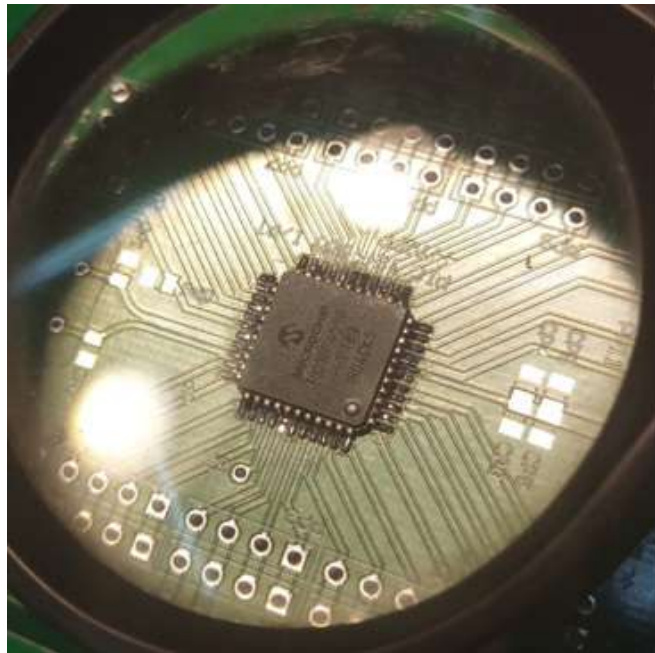


Figura 52: Soldado del PIC en VNA.

La placa completa luce de la siguiente manera:



Figura 53: Placa PCB del VNA. Con componentes.

En el anexo 9 de este documento se encuentran los esquemáticos de la placa y las tablas de todos

los componentes utilizados.

#### 4.7.3. Ficha Técnica

Se resumen en las siguientes tablas la ficha técnica de la placa VNA:

#### 4.7.4. Características eléctricas

Parameter	min	typ	max	Unit	Comments
Power Supply Voltage	4.9	5	5.1	V	Secciones PIC, ADF's y Banco de Mixers
Supply Current	-	310	-	mA	Ambos PLL (ADF's) enganchados
USB Voltage	-	5	-	V	Por cada USB
USB current	-	56	67	mA	Por cada USB
ADC CODEC Resolution	-	16	-	bits	<i>Sampling rate</i> a 44,1 kHz
UART Baud Rate	-	9600	-	baud	-

Tabla 4: Ficha Técnica: Características eléctricas.

#### 4.7.5. Características de RF

Parameter	min	typ	max	Unit	Comments
Characteristic Impedance	-	50	-	Ohm	Líneas Coplanares, Impedancia de conectores SMA ZL y Tx
RF out (ZL)	-35	-25	-15	dBm	Condición de adaptación: 50 Ohm en Puerto ZL
RF in (Tx)	-60	-25		dBm	Condición de adaptación: 50 Ohm en Puerto Tx
Frequency Range	300	-	3000	MHz	Puerto ZL como salida
Resolution	1	10	2700	MHz	-

Tabla 5: Ficha Técnica: Características de RF.

#### 4.7.6. Conectores e Interfaces

Nombre	Tipo	Conector	Descripción
5V-GND	-	Pin Headers	Alimentación de la placa
S11/S22	Out	USB tipo B hembra	Salida del Codec A
S21/S12	Out	USB tipo B hembra	Salida del Codec B
ZL	In/Out	SMA hembra	Conexión de Entrada del DUT
TX	In	SMA hembra	Conexión de Salida del DUT
UART	In/Out	Pin Headers	Conexión UART/USB

Tabla 6: Ficha Técnica: Interfaces.

## 5. Programa ejecutable: Software

El programa ejecutable del proyecto VNA tiene que cumplir las siguientes especificaciones:

- Interactuar con el usuario y permitirle configurar el barrido de medición.
- Conectarse mediante puerto USB/UART al microcontrolador PIC de la placa VNA y enviarle la frecuencia a generar.
- Adquirir las señales mediante la interfaz USB CODEC de la placa VNA.
- Procesar las señales efectuando un filtrado digital y calcular tanto amplitud como fase de cada señal.
- Construir el parámetro  $S$  correspondiente y almacenar los datos.
- Efectuar lógica de calibración para la eliminación de errores.
- Graficar los parámetros  $S$ .
- Abrir y guardar archivos de parámetros  $S$ .

La metodología de trabajo empleada para el diseño de Software se muestra en la figura 54. Para el diseño de la interfaz se utilizan las librerías de Python [26] en el entorno de Visual Studio Code [27]. Para el desarrollo de Backend y todas las funcionalidades del sistema se utiliza el entorno de Jupyter Notebook [28] (con Python), el cual facilita la configuración de barrido, las gráficas y las iteraciones de calibración.

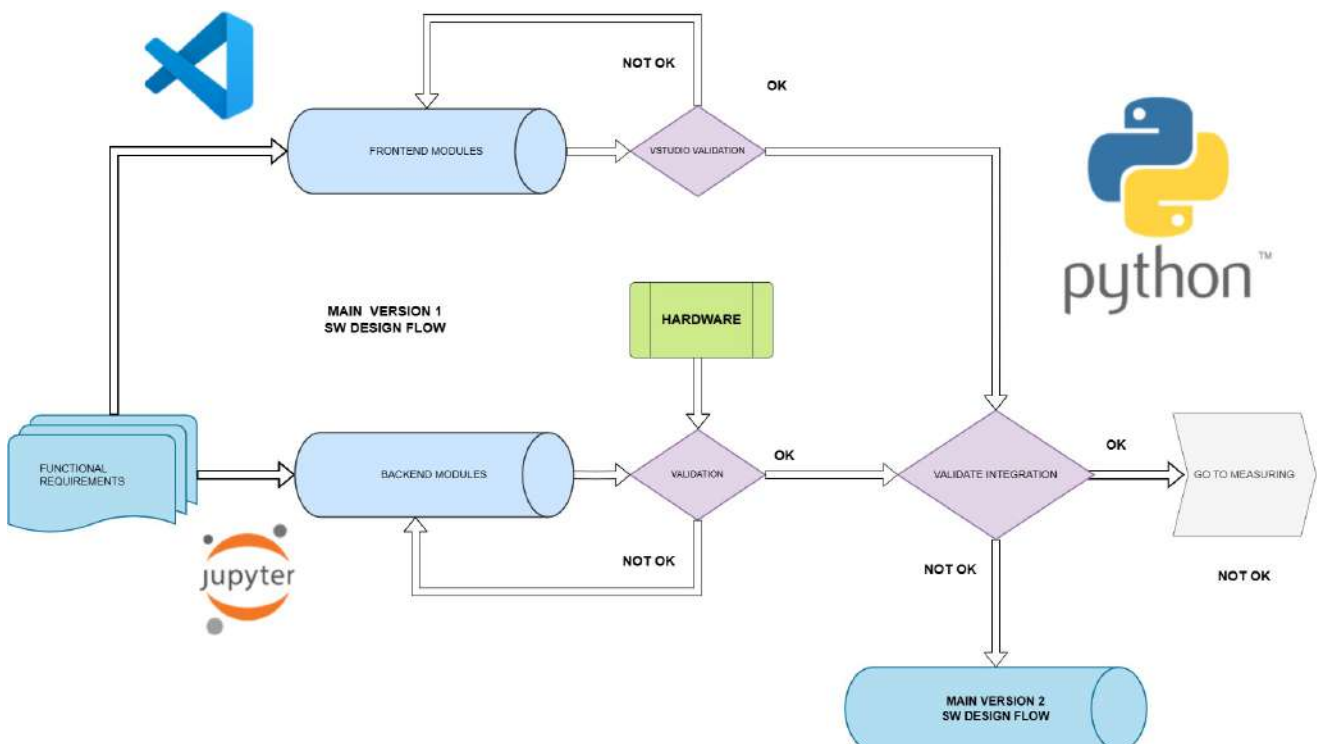


Figura 54: Metodología de Trabajo.

A partir de los requerimientos funcionales 3.3, se diseñan los bloques de Backend y Frontend a partir de diferentes funciones y librerías de Python. En paralelo, se prueban las funcionalidades del sistema que requieren tanto Software como Hardware, logrando así la validación de cada bloque. Finalmente, se integran las funcionalidades de Backend y Frontend en la interfaz final.

## 5.1. Interfaz de usuario (Frontend)

En la figura 55 se observa el diagrama de la interfaz de usuario o Frontend del Software:

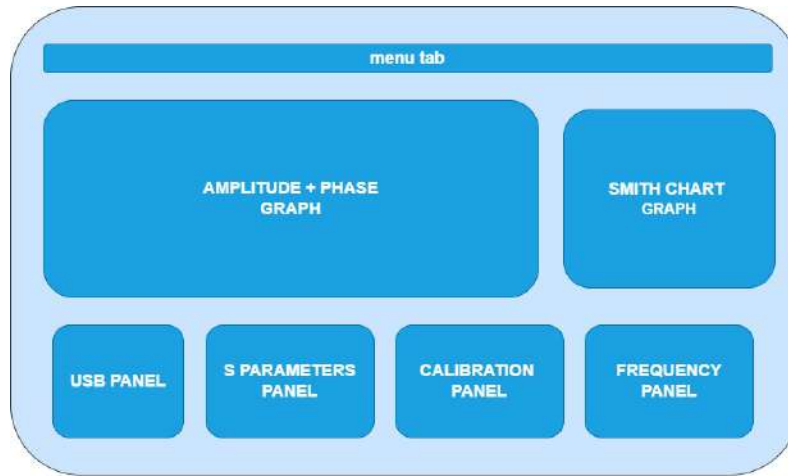


Figura 55: Diagrama de la interfaz de usuario.

En MENU TAB, el usuario es capaz de elegir las opciones típicas del procesamiento de archivos (New File, Open File, Save File). Para la visualización de los datos se muestran gráficas de amplitud y fase en función del rango de frecuencia y un diagrama de Smith para interpretar en términos de impedancia a los parámetros de reflexión  $S_{11}$  y  $S_{22}$ . En USB PANEL el usuario puede verificar la conexión de los puertos USB con la placa VNA (USB/UART y USB CODEC). En S PARAMETERS PANEL se elige el parámetro S a graficar (PLOT) o a medir (MEASURE) sin calibración. En CALIBRATION PANEL el usuario elige el parámetro de error correspondiente para medir (MEASURE), graficar (PLOT) pero también para efectuar la calibración correspondiente (CALIBRATE). Por último, en FREQUENCY PANEL, el usuario configura el rango de frecuencia a efectuar, tanto frecuencia inicial y final (FREQ INIT, FREQ END), el paso (FREQ STEP) y la cantidad de muestras por barrido (SAMPLES).

En la figura siguiente se observa el programa ejecutable ya desarrollado en Python, en conjunto con la librería de interfaces PyQt [29].

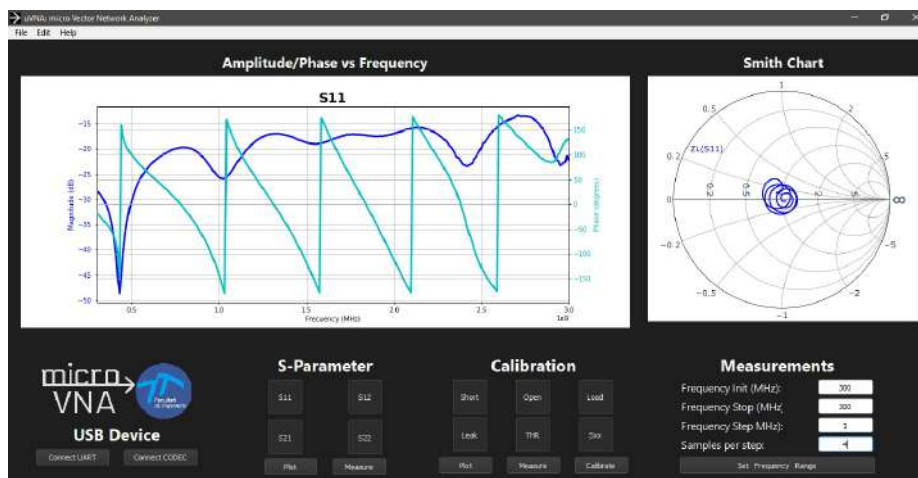


Figura 56: Frontend del Software: Interfaz de usuario.

## 5.2. Paneles y módulos (Backend)

Una vez inicializado el programa ejecutable, se ajustan los valores de frecuencia predeterminados y se calcula la cantidad de puntos  $n$  necesarios para generar vectores vacíos tanto para cada parámetro S como también para cada parámetro de error y otras variables auxiliares. Luego, se inicializa cada gráfica.

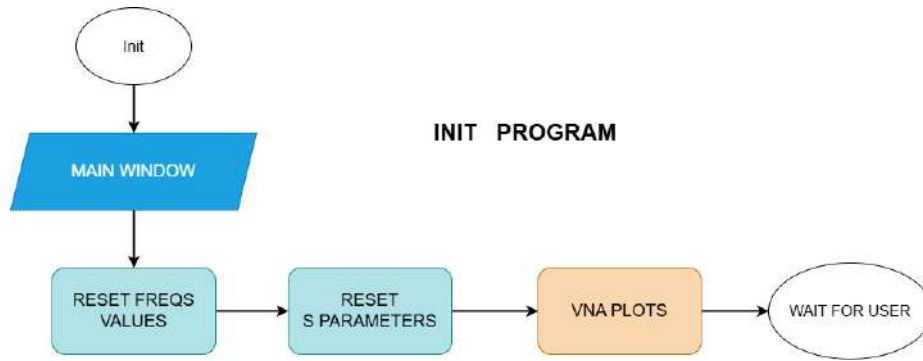


Figura 57: Inicialización del Software.

### 5.2.1. Menu Tab

En la sección superior del *menu tab* se encuentra el botón desplegable FILE, mediante el cual se podrá crear un nuevo archivo y borrar toda la información anterior, abrir un archivo de parámetros S para poder visualizar sus gráficas o bien guardar la información, ya sea de los parámetros medidos o bien desde algún archivo abierto anteriormente.

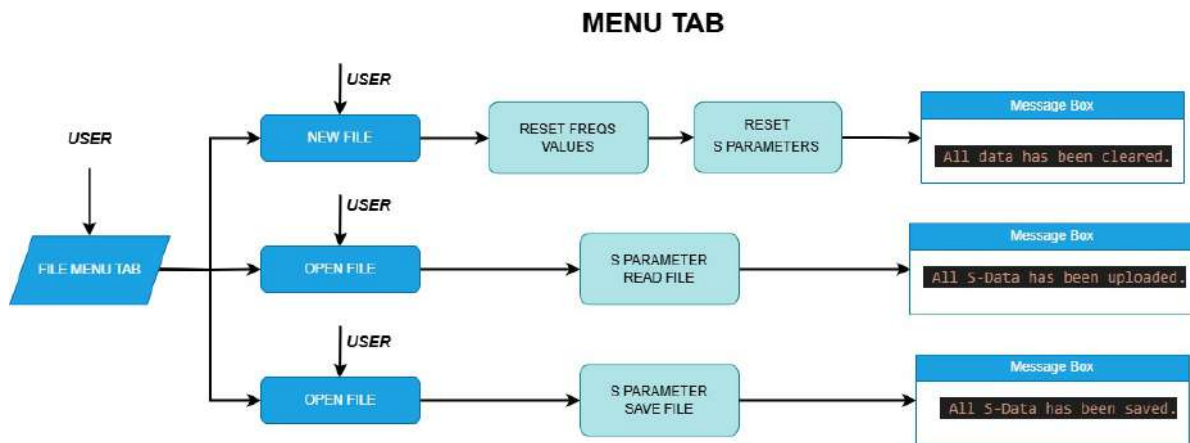


Figura 58: Menu Tab.

### 5.2.2. Panel de frecuencia

En este panel el usuario ingresa en formato texto con el teclado los valores de frecuencia para efectuar el barrido de medición. Los valores son: la frecuencia de inicio (Freq Init), la frecuencia final (Freq End), el paso entre frecuencias (Freq Step) y la cantidad de muestras por barrido (Samples) para efectuar un promediado de medición. Los valores de frecuencia deben verificarse tanto si algún valor no corresponde a valor numérico como para el caso de que esté por fuera del rango de UHF (300 MHz a 3GHz), es decir que debe cumplirse la siguiente relación:

$$300 \leq Freq_{init} < Freq_{End} \leq 3000 \tag{31}$$

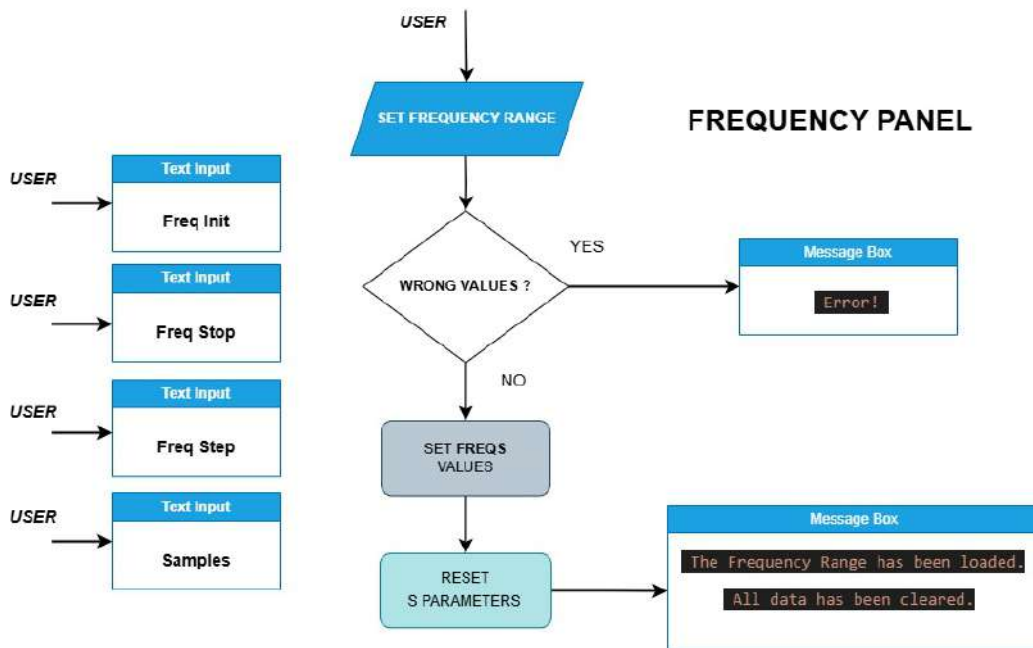


Figura 59: Panel de frecuencia.

Si los valores de frecuencias son válidos, se almacenan en el conjunto FREQS. Luego se borran los parámetros S anteriores.

### 5.2.3. Panel de conexión USB

Este panel tiene el propósito de verificar la conexión de cada puerto USB, asignando de manera automática mediante identificación de dispositivos al puerto virtual (COM14 por ejemplo) e inicializando las velocidades (9600 baud):

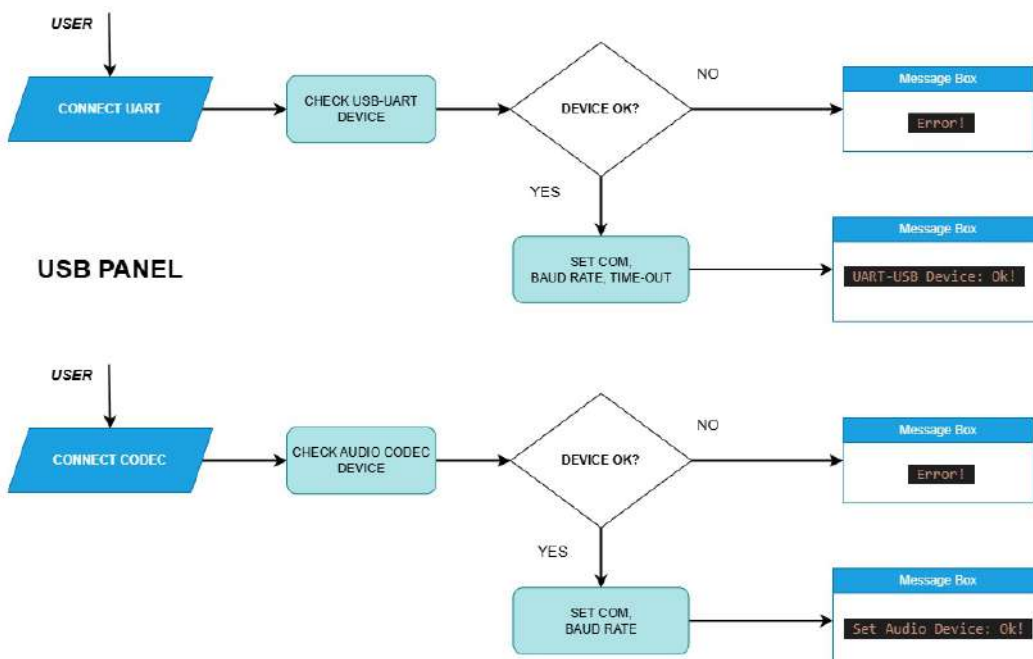


Figura 60: Panel de conexiones USB.

### 5.2.4. Panel de parámetros S

A partir de la selección de botones de alguno de los parámetros S (S11, S12, S21 o S22), se asigna a una variable auxiliar SS la cual será empleada para realizar tanto las funciones de graficación (VNA PLOTS) como de medición (en caso de apretar los botones PLOT o MEASURE, respectivamente).

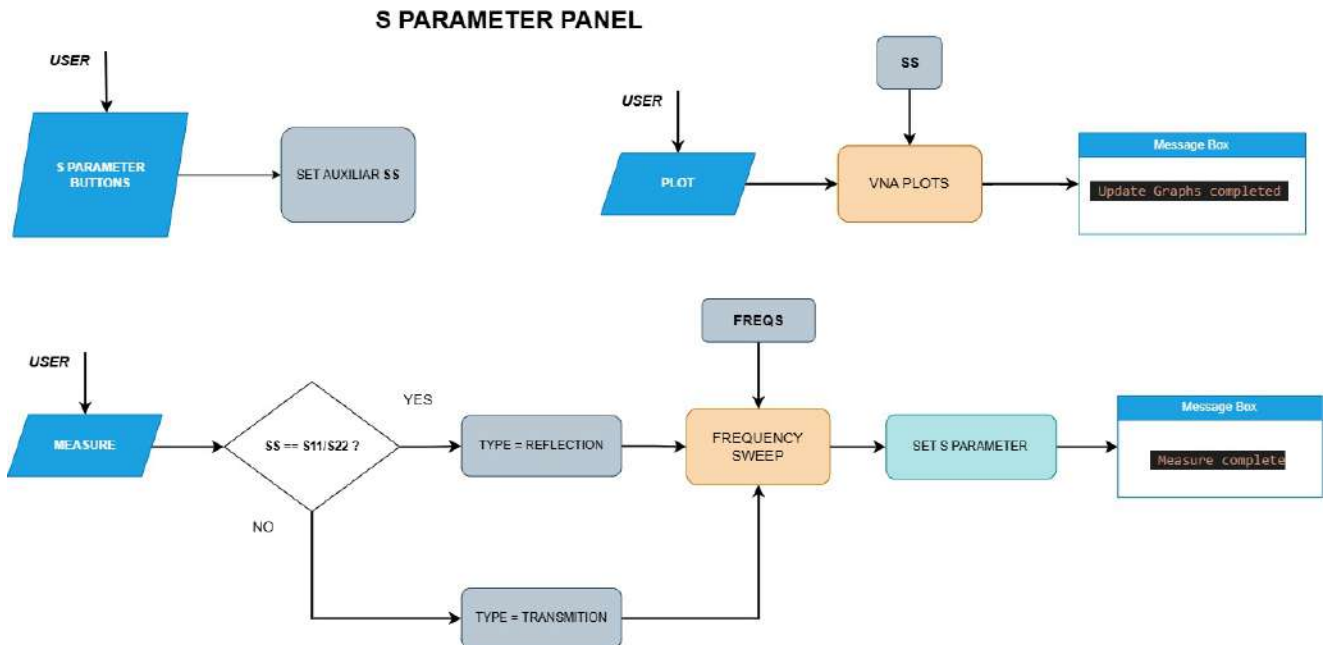


Figura 61: Panel de parámetros S.

Este panel logra que observar las gráficas sea intuitiva (PLOT), tanto en caso de abrir un archivo como en caso de haber realizado las mediciones, ya que estas últimas quedan en almacenamiento de las variables de parámetros S. Para la medición (MEASURE) correspondiente al tipo, ya sea de reflexión S11/S22 o de transmisión S12/S21, se llama a la función de barrido de frecuencia (*Frequency sweep*), la cual envía las frecuencias al PIC y luego adquiere y procesa las señales provenientes del CODEC. Estas mediciones se realizan sin tener en cuenta la calibración.

### 5.2.5. Panel de calibración

El panel de calibración es similar al anterior solo que agrega la lógica del tipo de calibración que emplea, ya sea de reflexión o de transmisión (Esto a partir de la variable  $SS=S_x$  del panel anterior). A partir del parámetro escogido se almacena en la variable auxiliar CAL para luego poder medir y almacenar (MEASURE) o graficar (PLOT).

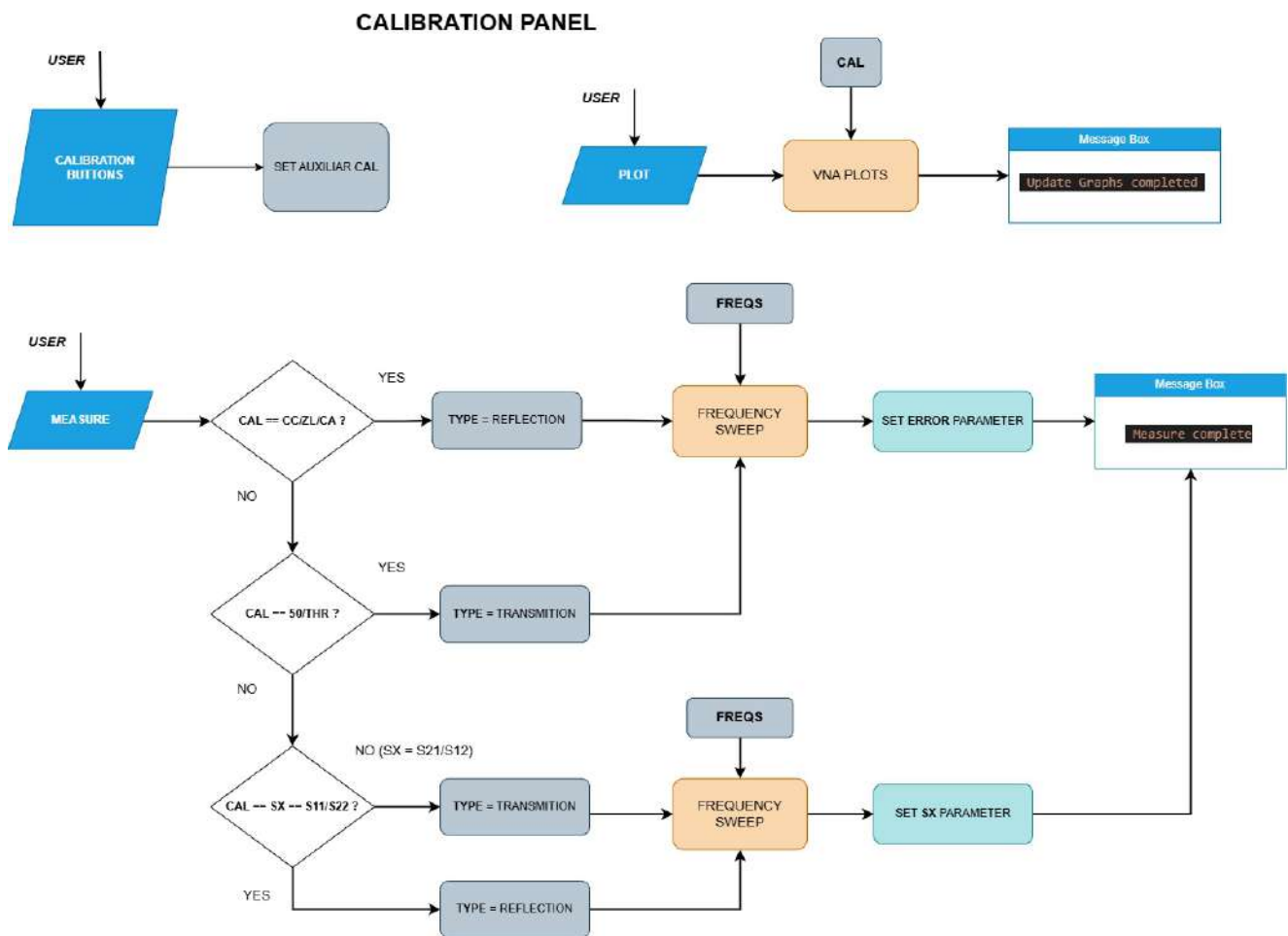


Figura 62: Panel de calibración.

Dependiendo del tipo de calibración (reflexión o transmisión) y las mediciones de carga, se aplica la función de corrección de reflexión (GAMMA CORRECTOR) o la función de corrección de transmisión (TRANSFER CORRECTOR):



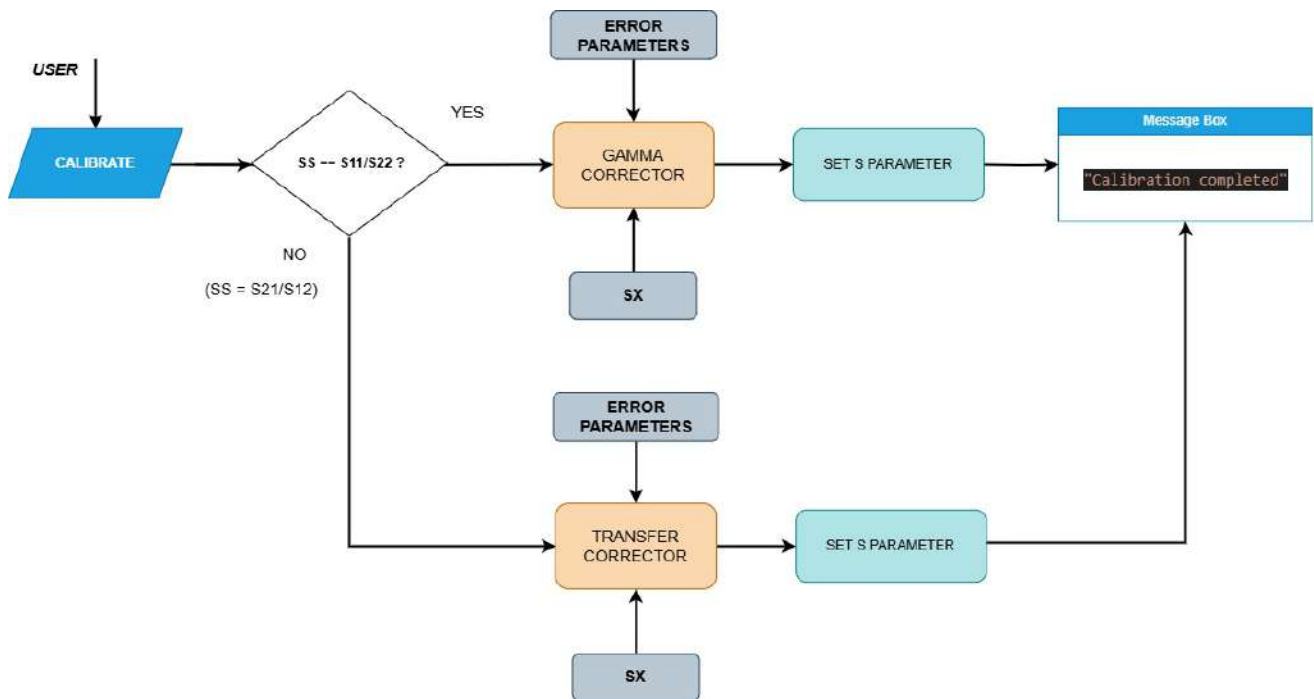


Figura 63: Panel de Calibración.

### 5.3. Algoritmos principales

Se presentan los principales algoritmos utilizados para realizar las funciones de Backend. Todos los códigos utilizados e implementados se encuentran en el Anexo 9.

#### 5.3.1. Barrido de frecuencias (Controlador)

El siguiente flujograma (figura 63) muestra la función de barrido, en la que se encarga de realizar las mediciones individuales para cada frecuencia del rango determinado por el usuario. Por cada valor, envía la frecuencia determinada o instantánea ( $F_{out}$ ) al PIC (SEND UART) y luego adquiere las señales, procesandolas y calculando el valor de S para almacenarlo en una variable auxiliar. Una vez terminado el barrido, suma cada parámetro S en una variable SUM, la cual irá almacenando cada parámetro de cada frecuencia por cada secuencia de barrido para luego realizar un promediado, según el valor de muestras (Samples) administrado por el usuario.

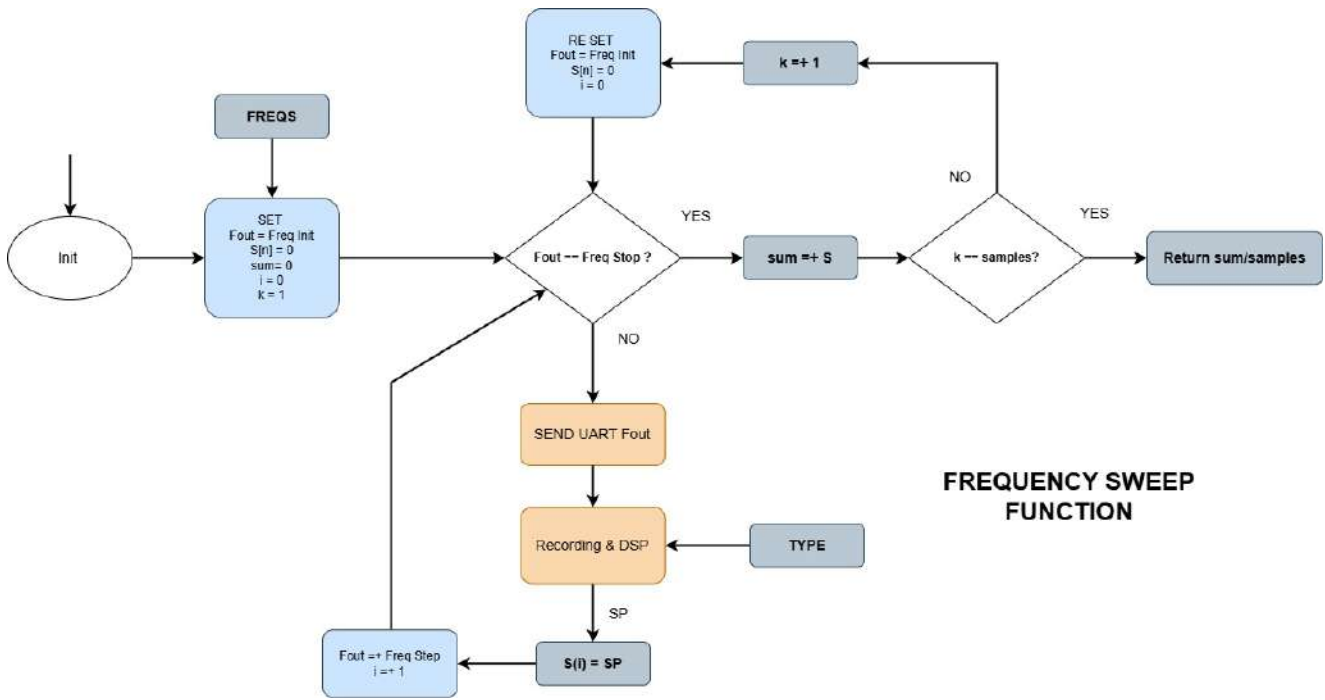


Figura 64: Función de barrido de frecuencias.

### 5.3.2. Envío de frecuencias (SEND UART)

A partir de la frecuencia instantánea  $F_{out}$ , la cual es un número entero entre 300 y 3000, se lo convierte en 2 bytes hexadecimales. Luego, se le agrega una cabecera de identificación ID (1 byte) y se envían los 3 bytes directamente al PIC.

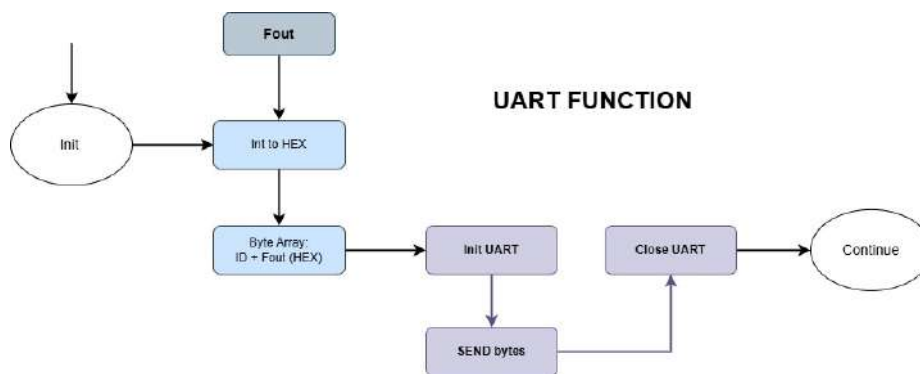


Figura 65: Envío de frecuencias.

### 5.3.3. Procesamiento de señales (DSP)

A partir del inicio de conexión con uno de los CODEC de la placa VNA, se reciben las señales digitalizadas en formato WAV con una duración de 0,1 segundos a una velocidad de muestreo de  $f_s = 44,1$  kHz. Dado que las señales de frecuencia intermedia son de 1 kHz con una resolución de 16 bits, eso implica tener aproximadamente 44 puntos por ciclo (período  $T$ ), con un total de 100 ciclos medidos, lo cual es más que suficiente para el cálculo de amplitud y fase de cada señal.

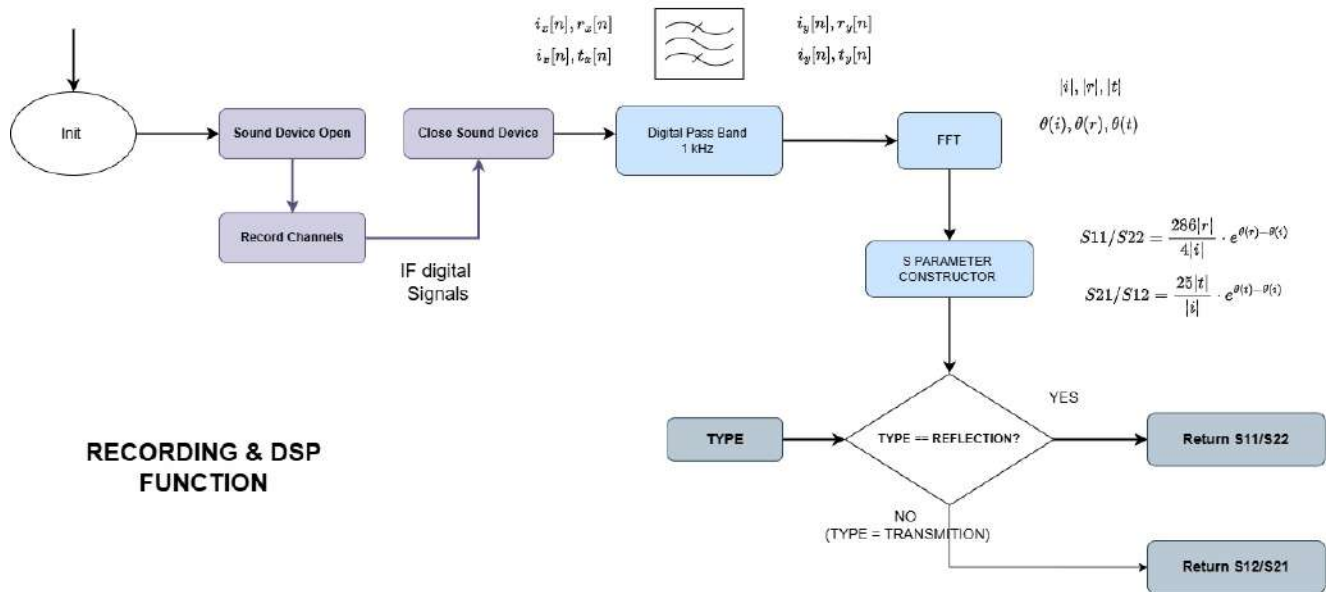


Figura 66: Procesamiento de señales.

Una vez adquiridas las señales y digitalizadas, pasan a través de un filtro pasa banda digital tipo Butterworth de orden 2  $H(z)$ , con frecuencia central 1 kHz y ancho de banda de 100 Hz. Esto es para eliminar las componentes indeseadas y obtener únicamente a la componente de IF de 1 kHz. Por otro lado, se emplea un método de filtrado hacia adelante y hacia atrás para eliminar el desfase producido por el filtro:

$$H(z) = \frac{5 \cdot 10^{-5} - 10^{-4} \cdot z^{-2} + 5 \cdot 10^{-5} \cdot z^{-4}}{1 - 3,94 \cdot z^{-1} + 5,86 \cdot z^{-2} - 3,9 \cdot z^{-3} + 0,98 \cdot z^{-4}} \quad (32)$$

Una vez filtradas las señales se aplica la Transformada Rápida de Fourier (FFT) [30] a cada una de ellas:

$$\text{fft\_signal}[k] = \sum_{n=0}^{N-1} \text{signal}[n] \cdot e^{-j \frac{2\pi}{N} kn} \quad (33)$$

donde  $N = T \cdot f_s$  es el número de muestras. Teóricamente, la FFT se puede considerar como una sumatoria de componentes de índices  $k$ :

$$f[k] = \frac{k \cdot f_s}{N} \quad (34)$$

La frecuencia objetivo es 1000 Hz, por lo que, teniendo en cuenta una tolerancia de 50 Hz (frecuencia de línea):

$$1000 \text{ Hz} - 50 \text{ Hz} \leq f[k] \leq 1000 \text{ Hz} + 50 \text{ Hz} \quad (35)$$

De esta forma, la amplitud de la componente buscada es:

$$\text{amp}[k] = \frac{2}{N} \sqrt{\text{Re}(\text{fft\_signal}[k])^2 + \text{Im}(\text{fft\_signal}[k])^2} \quad (36)$$

y la fase:

$$\text{phase}[k] = -\text{arctg}\left(\frac{\text{Im}(\text{fft\_signal}[k])}{\text{Re}(\text{fft\_signal}[k])}\right) \quad (37)$$

Es decir que, a partir del algoritmo de la FFT se calculan las amplitudes  $|i_x|$ ,  $|r_x|$  y  $|t_x|$  y las fases iniciales  $\theta(i_x)$ ,  $\theta(r_x)$  y  $\theta(t_x)$ . A partir de estos valores y las ecuaciones desarrolladas en 4.6.1 y 4.6.2, los parámetros de reflexión y transmisión se calculan como:

$$\Gamma(\text{Reflection}) = \frac{286}{4} \frac{|r_x|}{|i_x|} \cdot e^{\theta(r_x) - \theta(i_x)} \quad (38)$$

$$G(\text{Transfer}) = 25 \frac{|t_x|}{|i_x|} \cdot e^{\theta(t_x) - \theta(i_x)} \quad (39)$$

Luego, para cada medición, estos valores se almacenan en la variable correspondiente. Para obtener los valores reales del DUT se aplican los métodos de calibración que se detalla en la siguiente subsección.

## 5.4. Calibración

### 5.4.1. Calibración en reflexión (GAMMA CORRECTOR)

Para la reflexión se emplea el método de calibración SOL (Short-Open-Load) [31], la cual ajusta la medición de reflexión ya sea S11 o S22 teniendo en cuenta 3 mediciones en el puerto de salida del VNA (Entrada a 3 DUT's de calibración): Corto Circuito (CC), 50  $\Omega$  (ZL), y Circuito Abierto (CA).

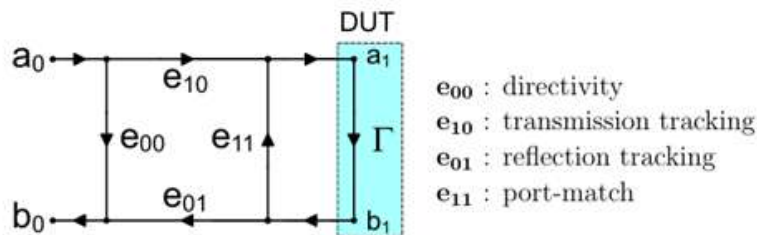


Figure 5: One-Port Error Model

Figura 67: Modelo de 1 puerto: calibración SOL. Fuente: [31].

Donde el coeficiente de reflexión medido sin calibrar es:

$$\Gamma_M = \frac{b_0}{a_0} = \frac{e_{00} - \Delta_e \Gamma}{1 - e_{11} \Gamma} \quad (40)$$

El determinante de error es:

$$\Delta_e = e_{00}e_{11} - (e_{10}e_{01}) \quad (41)$$

Por lo que el coeficiente de reflexión calibrado es:

$$\Gamma = \frac{\Gamma_M - e_{00}}{\Gamma_M e_{11} - \Delta_e} \quad (42)$$

Este método de calibración logra corregir los errores debidos a desadaptaciones, pérdidas de directividad, pérdidas debida a conectores, etc. El procedimiento es el siguiente:

1. Conectar Puerto ZL (VNA) a carga Corto Circuito. Presionar botón CC y medir con MEASURE. Generar vector de carga  $GM_{CC}$ .

2. Conectar Puerto ZL (VNA) a carga  $50\Omega$ . Presionar botón ZL y medir con MEASURE. Generar vector de carga  $GM_{50}$ .
3. Conectar Puerto ZL (VNA) a carga Circuito Abierto. Presionar botón CA y medir con MEASURE. Generar vector de carga  $GM_{CA}$ .
4. Conectar Puerto ZL (VNA) al DUT deseado. Presionar botón Sx y medir con MEASURE. Generar vector  $S_{11}$  o  $S_{22}$ .
5. Presionar CALIBRATE para calibrar. Luego, PLOT para graficar el parámetro medido y corregido.

A partir de los vectores de error, se arma un sistema de ecuaciones (ya que se conoce para cada carga cual sería el  $\Gamma$  real):

$$\begin{cases} GM_{CC} = e_{00} - GM_{CC} \cdot e_{11} + \Delta_e & (a) \\ GM_{50} = e_{00} & (b) \\ GM_{CA} = e_{00} + GM_{CA} \cdot e_{11} - \Delta_e & (c) \end{cases} \quad (43)$$

Al tratarse de un sistema lineal de 3 ecuaciones con 3 coeficientes complejos incógnita, se calculan estos coeficientes de error para luego determinar el valor real de  $\Gamma$ , es decir, la función de corrección de reflexión:

$$\Gamma = \frac{\Gamma_M - e_{00}}{\Gamma_M e_{11} - \Delta_e} \quad (44)$$

### 5.4.2. Calibración en transmisión (TRANSFER CORRECTOR)

Para la transmisión se emplea el método de calibración THR (*THROUGH*) [31], la cual ajusta la medición de transmisión ya sea  $S_{21}$  o  $S_{12}$  teniendo en cuenta 2 mediciones en el puerto de entrada del VNA (Salida a 3 DUT de calibración o cable de conexiones): Puertos a  $50\Omega$  y Cable de interconexión (Salida VNA - Entrada VNA).

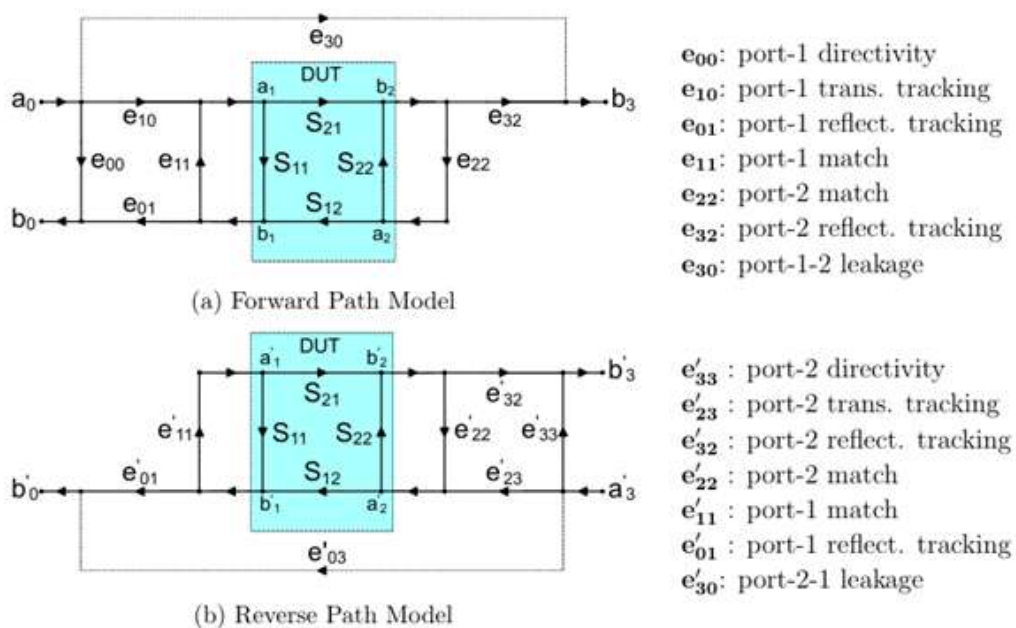


Figura 68: Modelos Forward ( $S_{21}$ ) y Reverse ( $S_{12}$ ): calibración THR. Fuente: [31].

Este método de calibración es un poco más complejo ya que logra corregir otros tipos de errores, como pueden ser las múltiples reflexiones, desadaptaciones, fugas de señal, pérdidas de directividad, pérdidas debidas a conectores, etc. Despreciando los errores debido a la reflexión en el puerto de salida del DUT, el procedimiento a seguir:

1. Conectar Puertos ZL y Tx (VNA) a carga  $50\Omega$ . Presionar botón LEAK y medir con MEASURE. Generar vector de error  $e_{30}$ .
2. Interconectar Puertos ZL y Tx (VNA) con el cable con el que se interconectaría el DUT. Presionar botón THR y medir con MEASURE. Generar vector de error  $e_{1032}$ .
3. Conectar el DUT con el cable anterior. Presionar botón Sx y medir con MEASURE. Generar vector  $S_{21}$  o  $S_{12}$ .
4. Presionar CALIBRATE para calibrar. Luego, PLOT para graficar el parámetro medido y corregido.

A partir de los vectores de error y la medición sin calibrar  $G_M$ , la corrección de transferencia es simplemente:

$$G_{DUT} = \frac{G_M - e_{30}}{e_{1032}} \quad (45)$$

## 6. Plan de pruebas

Esta sección corresponde al plan de pruebas del proyecto VNA y tiene el propósito de proveer información acerca de la realización de diversas pruebas para verificar el funcionamiento del sistema completo (Firmware + Hardware + Software) y validar si cumple con los requerimientos y/o funcionalidades.

Se encuentra dirigido hacia:

- Desarrollador del dispositivo y/o sistema.
- Cátedra de Trabajo Final.
- Estudiantes y/o investigadores del Laboratorio de Comunicaciones (LAC-ICYTE-UNMDP).

Se presenta el plan de pruebas desarrollado para efectuar a lo largo de este proyecto, las cuales se puede clasificar en:

- Pruebas unitarias: se realizan sobre una sola etapa o componente.
- Pruebas integrales: se realizan sobre distintos módulos o componentes al momento de integrarlos entre sí.
- Pruebas de homologación: se realizan sobre todo el sistema completo para validar requerimientos.
- Pruebas de aceptación: se realizan sobre el producto final para verificar funcionalidades del sistema.

### 6.1. Ambientes de prueba

Los ambientes de prueba son los determinados por cada componente, etapa o integración de varias, a los cuales se le realizan pruebas para validar su funcionalidad y/o requerimiento:

- MPLAB X IDE [14] (Embedded C)+ Microcontrolador PIC [11].
- MPLAB X IDE (Embedded C)+ Microcontrolador PIC + Generadores RF [16].
- Generadores RF + Receptor Heterodino.
- Visual Studio Code (Python) [27].
- CODECs [24] (USB) + Visual Studio Code (Python).
- Generadores RF + Receptor Heterodino + Codecs (USB) + Visual Studio Code (Python).
- Visual Studio Code (Python) + Microcontrolador PIC + Generadores RF + MPLAB X IDE (Embedded C).
- Sistema Completo: Visual Studio Code (Python) + USB/UART + Microcontrolador PIC + Generadores RF + Receptor Heterodino + Codecs (USB) + Visual Studio Code (Python).

## 6.2. Recursos, herramientas e instrumentos

Para llevar a cabo este plan de pruebas se utilizan las siguientes herramientas e instrumental, el cual es proporcionado por el Laboratorio de Comunicaciones (LAC) de la Facultad de Ingeniería de la Universidad Nacional de Mar del Plata.

- Computadora personal (PC).
- Programador PICKit V2 [15].
- Fuente de alimentación regulable: Kenwood PA36-3A [32].
- Conversor USB/UART: PI232 TTL-UART [12].
- Analizador de espectro: Rigol RSA3030N (VNA incluido) [2].
- Osciloscopio: Tektronix TDS 210 [33].
- Generador de señales: HP 33120A [34].
- Atenuador 110dB: HP 8496B [35].
- Multímetro: UNIT UNIT-55 [36].
- Cargas estándares: 50  $\Omega$ , circuito abierto, corto circuito.
- Cables de RF y conectores varios.

## 6.3. Políticas de trabajo

El plan de pruebas se ejecuta en su totalidad en el Laboratorio de Comunicaciones en un entorno de laboratorio no controlado a interferencias electromagnéticas provenientes de otros equipos.

Los resultados obtenidos durante las pruebas realizadas se reportan directamente a los directores del proyecto: el Dr. Ing. Ramiro Avalos Ribas y al Ing. Brian Gluzman, quienes acompañan y supervisan todo el proceso de las pruebas.

## 6.4. Pruebas de verificación

Se enumeran las pruebas de verificación unitarias e integrales:

1. Desarrollo de Firmware en PIC.
2. Generación de señales de RF.
3. Recepción de señales de RF.
4. Adquisición de señales de audio.
5. Barrido de frecuencias.
6. Procesamiento de señales y parámetros S.



### 6.4.1. Desarrollo de Firmware en PIC

Para esta prueba unitaria se desarrolla un código de prueba del Firmware en el Software MPLAB IDE (en PC), para luego compilarlo y cargar mediante el PICKITV2 hacia el microcontrolador PIC de la placa, mediante la comunicación ICSP. El programa puede ser el parpadeo de los LEDs del microcontrolador. Es de esperar que la compilación sea exitosa.

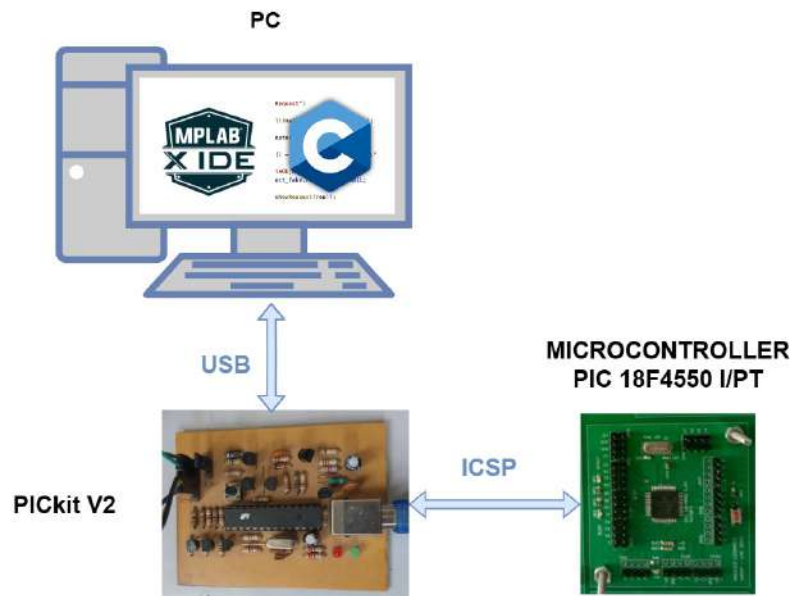


Figura 69: Banco de pruebas: compilación de firmware.

### 6.4.2. Generación de señales de RF

Se debe escribir el código del Firmware correspondiente a la carga de registros para cada comunicación SPI, es decir, un conjunto de registros por generador, a una potencia de salida de  $+5dBm$  y frecuencias  $f_1$  y  $f_1 + \Delta f$ . Una vez compilado y cargado el programa en PIC, se colocan los cables de comunicación SPI del PIC hacia los generadores ADF4351.

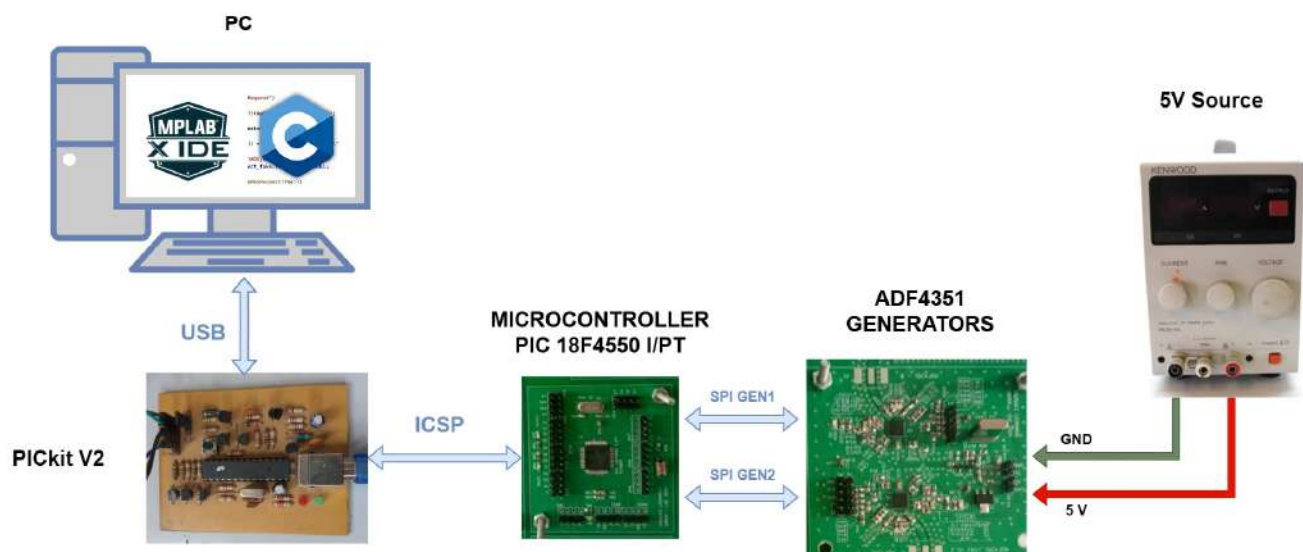


Figura 70: Banco de pruebas: carga de registros.

A la par de esto, se debe encender la sección de RF con una fuente de 5V 1 A. Para probar el

funcionamiento del regulador se utiliza un multímetro. La tensión que debe llegar a los generadores debe ser de 3,3 V. Para verificar la señal de referencia de 10 MHz hacia los generadores se emplea un osciloscopio.

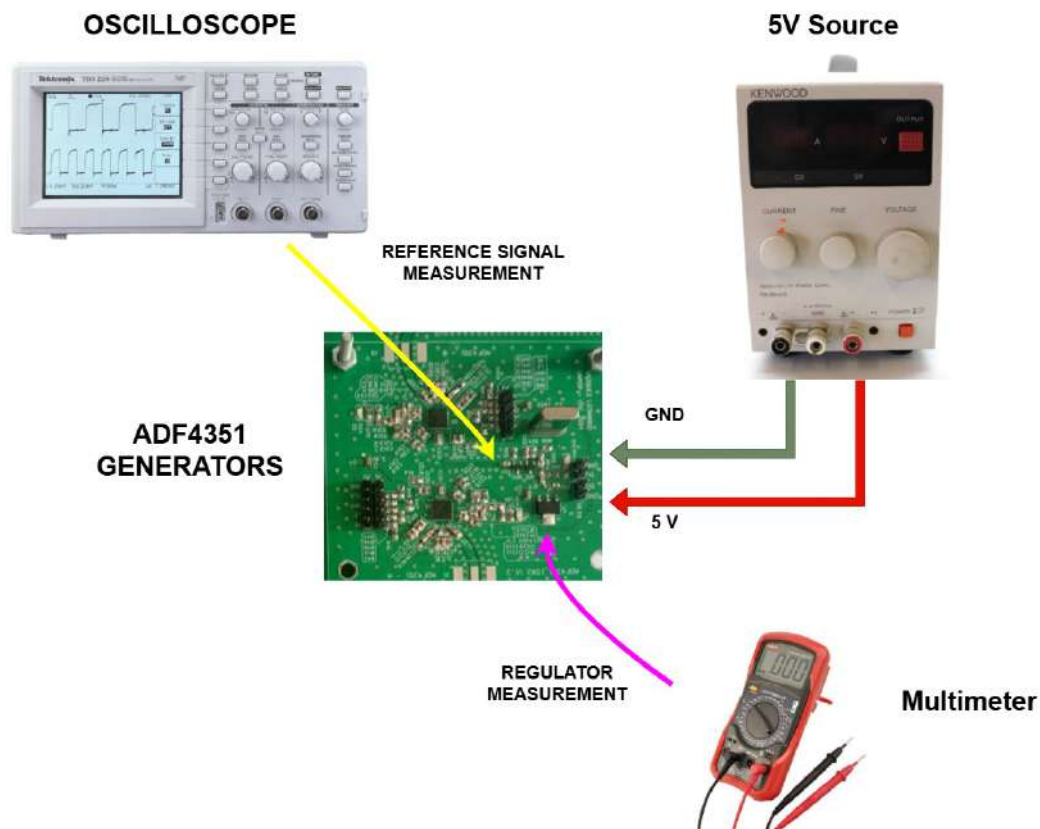


Figura 71: Banco de pruebas: sección de generadores de RF.

Luego, reiniciar el PIC y verificar LEDs de enganche de cada generador. El siguiente paso consiste en verificar potencia y frecuencia de las señales generadas, una por generador. Para ello conectar la salida de la placa con los conectores ZL y TX, ambos al Analizador de espectro, de manera individual. Medir frecuencias y potencias. Desde el puerto ZL se mide la potencia de salida de la placa. Desde el puerto TX se mide el acoplamiento desde el oscilador local hacia la entrada de RF de unos de los Mixers.

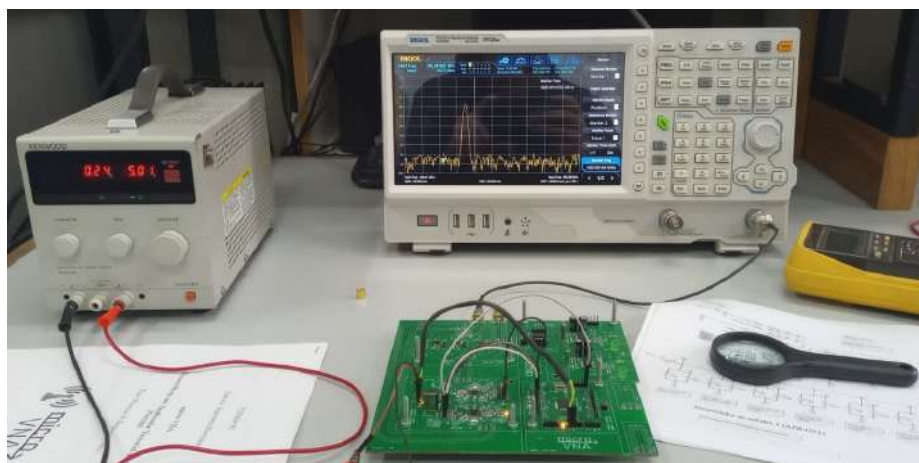


Figura 72: Banco de pruebas: mediciones de RF.

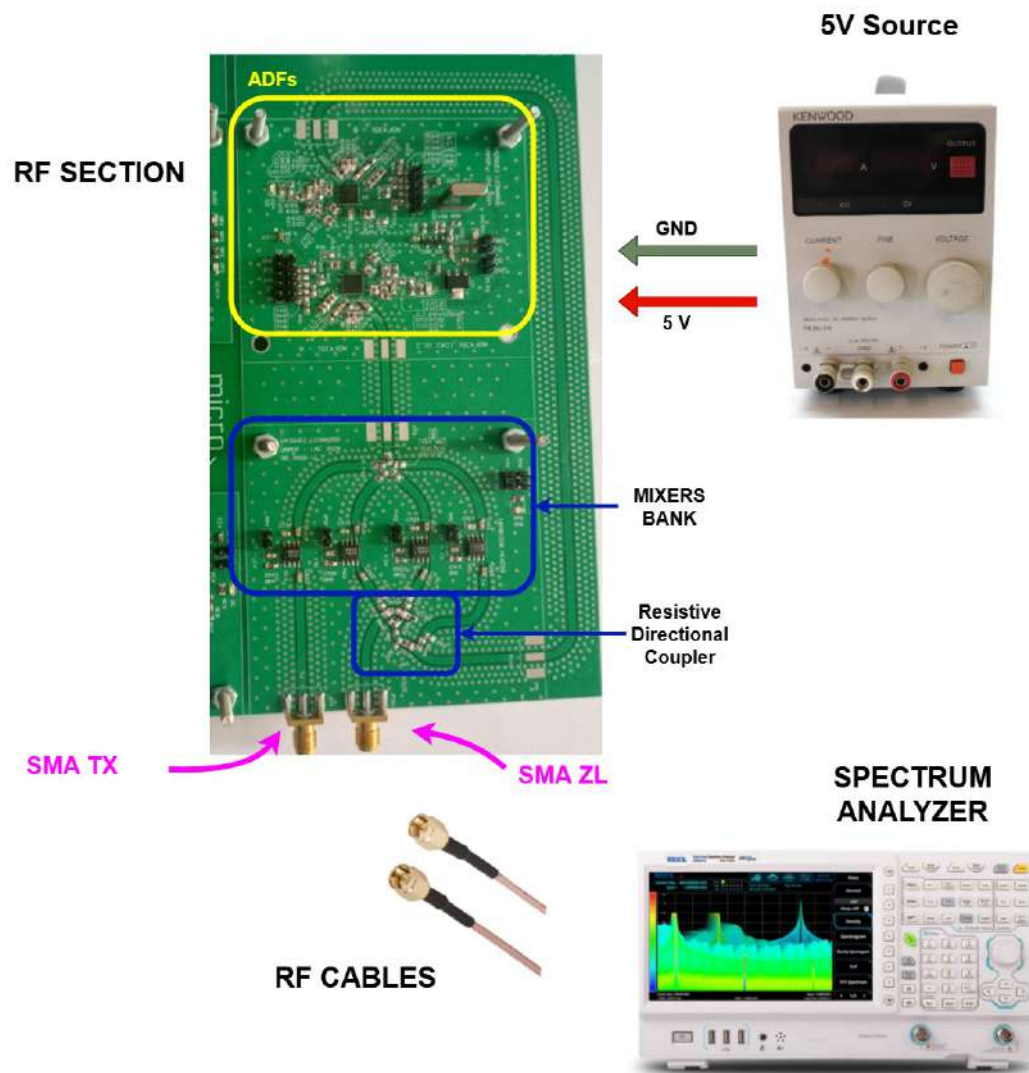


Figura 73: Banco de pruebas: mediciones de RF.

Repetir estas pruebas para frecuencias diferentes para así verificar la generación de señales a lo largo de la banda UHF.

#### 6.4.3. Recepción de señales de RF

Para verificar funcionamiento de la etapa del receptor heterodino, es decir, el banco de Mixers, se debe encender el equipo y medir con el osciloscopio las señales de salida de estos últimos, tanto  $i_x$  como  $t_x$ ,  $r1_x$  y  $r2_x$ . En principio, se debe asegurar que la diferencia de frecuencias en la salida de los Mixers sea  $\Delta f$ . Luego, colocar diferentes cargas en el conector ZL para las mediciones de reflexión. Para cada carga, las salidas de los Mixers (incidente y reflejada) deben presentar comportamientos diferentes y acordes a la característica del coeficiente de reflexión. Por otro lado, se emplea un cable de RF entre ZL y TX para la medición de transmisión. La diferencia entre la señal de incidencia y transmisión debe ser de 19,1dB, como teóricamente predice el acoplador resistivo direccional.

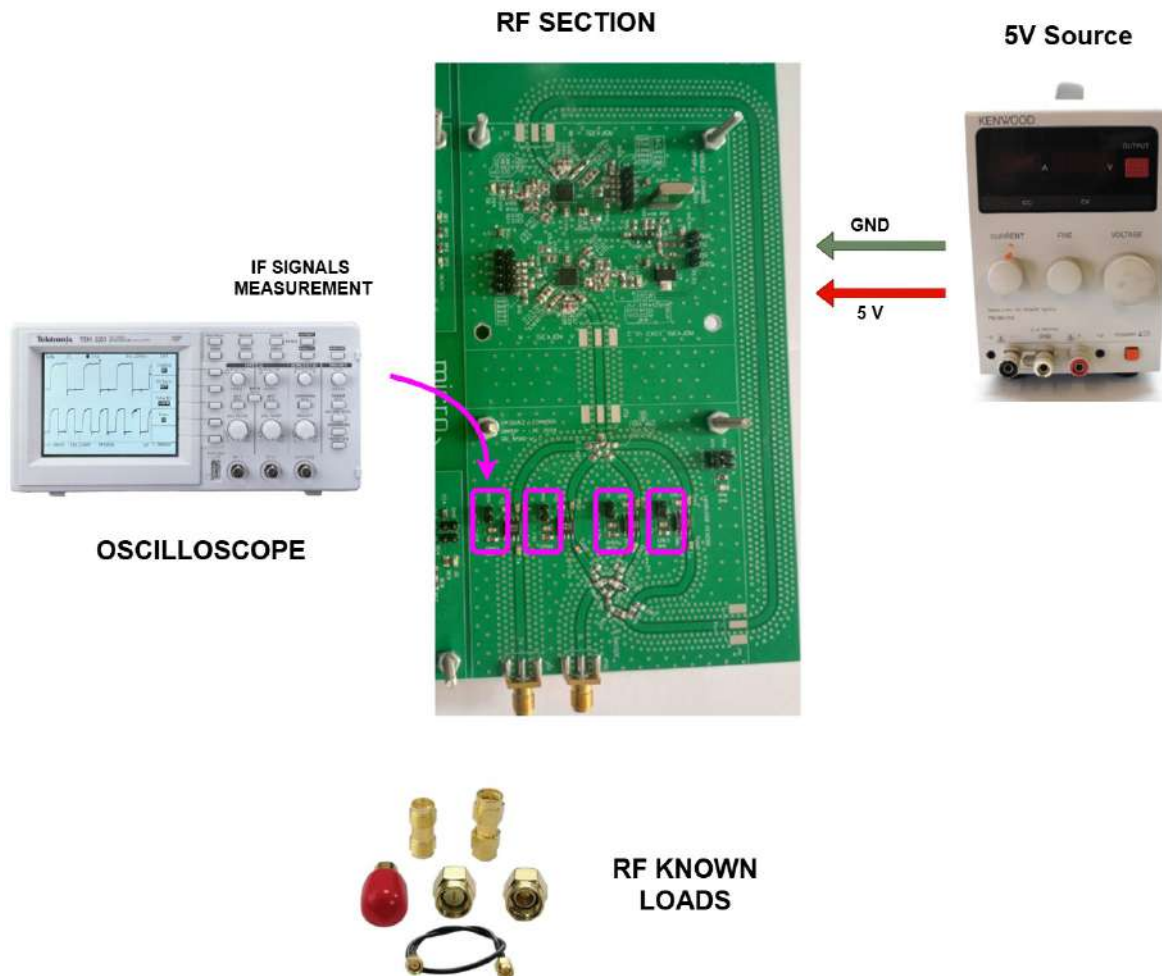


Figura 74: Banco de pruebas: mediciones de IF.

#### 6.4.4. Adquisición de señales

En esta etapa se prueba el funcionamiento de los CODECs de audio, tanto de reflexión (S11 o S22) como de transmisión (S12 o S21). Para ello se emplea un generador de señales para poder configurar las características de las señales tanto en amplitud como en frecuencia, para cada canal de cada CODEC de audio. Al mismo tiempo, se prueba la comunicación USB con el Software Backend en PC para la adquisición y procesamiento de las señales, verificando cálculos de amplitud y fase y la optimización del filtro digital.

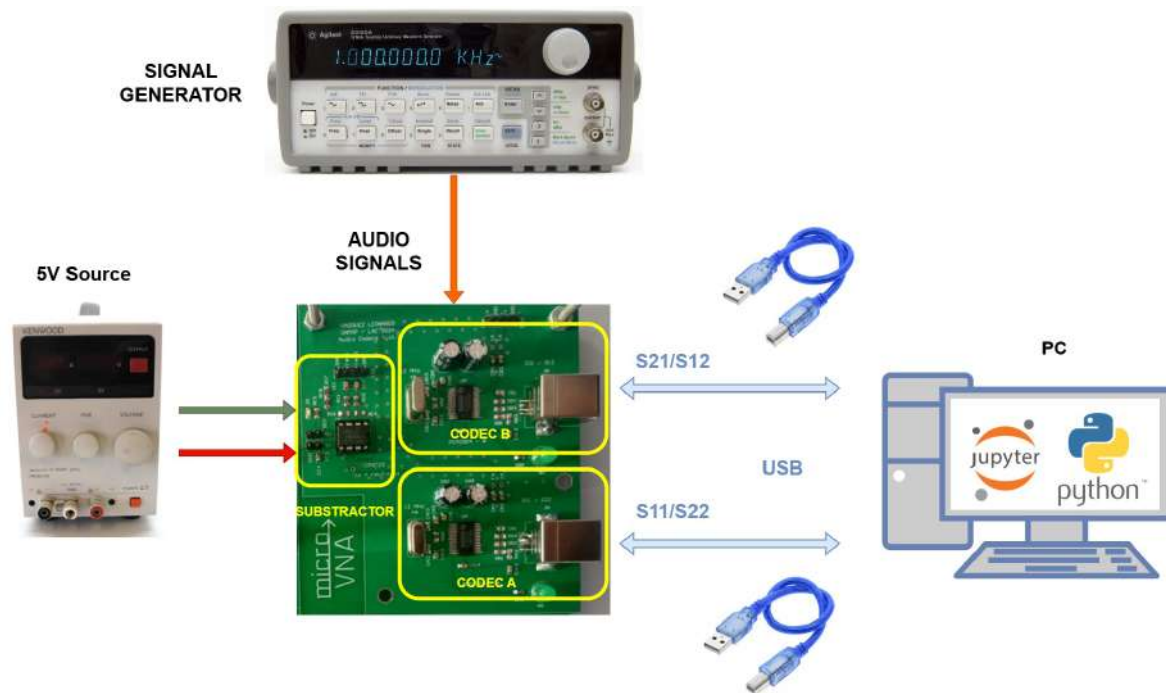


Figura 75: Banco de pruebas: mediciones de adquisición.

#### 6.4.5. Barrido de frecuencias

El último paso consiste en probar el funcionamiento del controlador en Software, dedicado al barrido de frecuencias, sincronizando el envío de las mismas hasta la recepción y grabación de las señales.

En primer lugar, se verifica el cálculo de registros del PIC, modificando el Firmware de este para el envío de los 6 registros de 32 bits cada uno (por cada frecuencia). Se compara entonces los registros calculados por el PIC y los calculados en PC.

La siguiente prueba es verificar la recepción de frecuencias desde la PC hacia el PIC vía el convertor UART/USB. Desde el algoritmo de barrido en PC envía frecuencia por frecuencia dentro de un rango cualquiera de UHF (paso mínimo de 1 MHz). Verificar que el PIC reciba correctamente los bytes de frecuencia reenviándolos hacia la PC.

Por último se prueba la carga de registros en cada uno de los generadores. Se encienden los generadores y se envían frecuencias desde PC al PIC, inicializando el barrido. Verificar LEDs de enganche para cada frecuencia (realizar barrido manual o lento). Medir salida ZL y entrada TX de la placa con analizador de espectro. Verificar barrido.

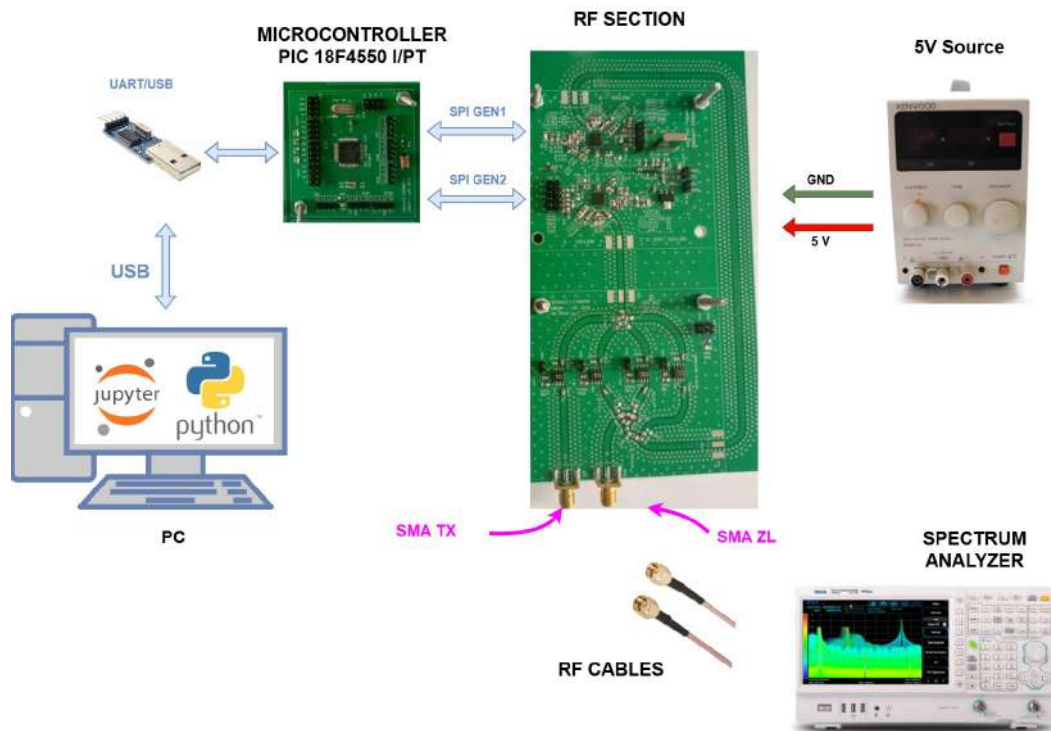


Figura 76: Banco de pruebas: mediciones de barrido de RF.

#### 6.4.6. Procesamiento de señales y parámetros S

Una vez corroborado el funcionamiento de la adquisición, se prueba el sistema completo conectando la parte de recepción de RF con los CODECs de audio. Se inicializa el barrido para así calcular los parámetros S deseados.

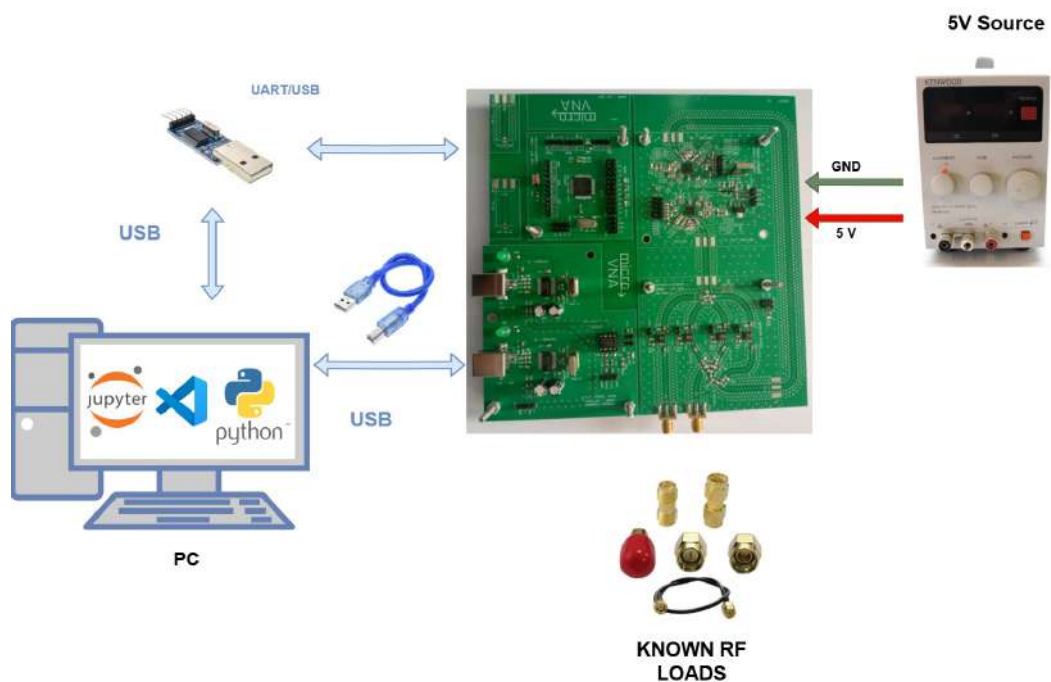


Figura 77: Banco de pruebas: sistema completo.

## 7. Mediciones y resultados

Para homologar el funcionamiento del sistema y medir los parámetros S de un DUT cualquiera, es necesario realizar las pruebas de mediciones sobre cargas conocidas y de calibración:

1. Pruebas de calibración en reflexión
2. Pruebas de calibración en transmisión

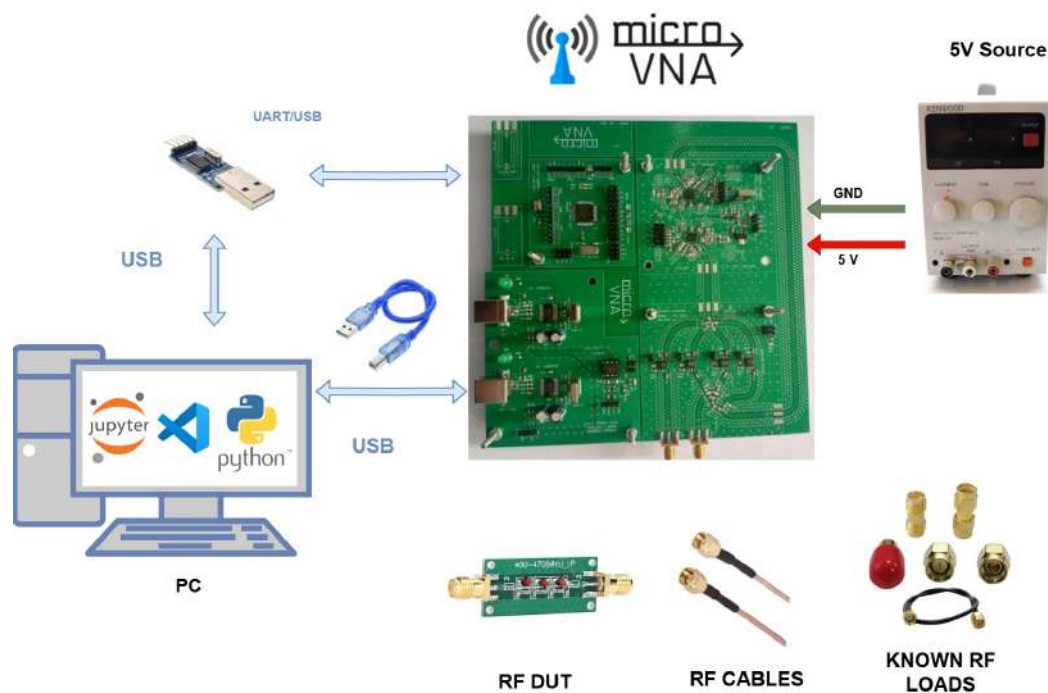


Figura 78: Banco de medición.

No obstante, se limita el ancho de banda del sistema a 2100 MHz como máximo debido a comportamientos no deseados en las señales medidas para valores superiores de frecuencias, lo cual se debe principalmente a las pérdidas del material FR4 y al comportamiento no lineal de los Mixers IAM81008 por encima de los 2 GHz. También se descartan aquellas frecuencias puntuales que no hayan podido ser generadas desde la sección de RF de los ADF4351.

### 7.1. Pruebas de calibración

#### 7.1.1. Reflexión

Para inicializar estas pruebas se conecta el USB Codec S11/S22. Luego, se ajusta un barrido deseado dentro de UHF, el cual por ejemplo puede ser entre 1500 MHz y 2100 MHz en pasos de 50 MHz. Para cada carga estándar en puerto ZL, medir y adquirir las señales de reflectometría según los siguientes pasos, para así obtener los coeficientes de error, generar la matriz de calibración y medir otra vez las mismas cargas estándares como un DUT deseado.

1. Conectar Puerto ZL (VNA) a carga Corto Circuito. Presionar botón CC y medir con MEASURE. Generar vector de carga  $GM_{CC}$ .
2. Conectar Puerto ZL (VNA) a carga 50  $\Omega$ . Presionar botón ZL y medir con MEASURE. Generar vector de carga  $GM_{50}$ .

3. Conectar Puerto ZL (VNA) a carga Circuito Abierto. Presionar botón CA y medir con MEASURE. Generar vector de carga  $GM_{CA}$ .
4. Conectar Puerto ZL (VNA) al DUT deseado. Presionar botón Sx y medir con MEASURE. Generar vector  $S_{11}$  o  $S_{22}$ .
5. Presionar CALIBRATE para calibrar. Luego, PLOT para graficar el parámetro medido y corregido.

Una vez realizada la calibración, se vuelven a medir tales cargas para homologar la mediciones:

### Carga Corto Circuito

Se espera que esta medición sea 0 dB en magnitud y  $180^\circ$  en fase, con los puntos de medición en el extremo izquierdo del ábaco de Smith:

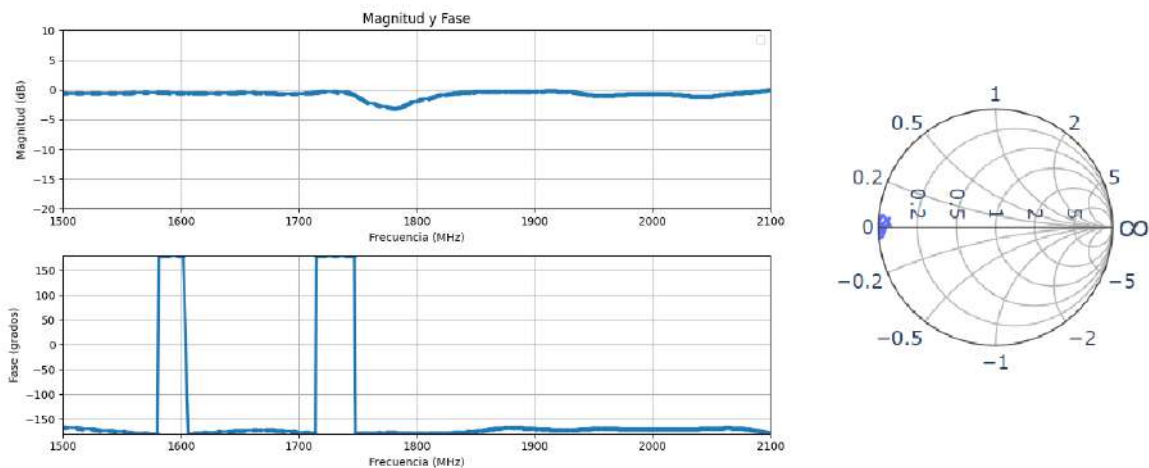


Figura 79: Medición de Carga Corto Circuito. Magnitud y fase.

### Carga 50 Ohm

Se espera que esta medición sea menor a  $-20$  dB en magnitud y con fase despreciable, con los puntos de medición en el centro del ábaco de Smith:

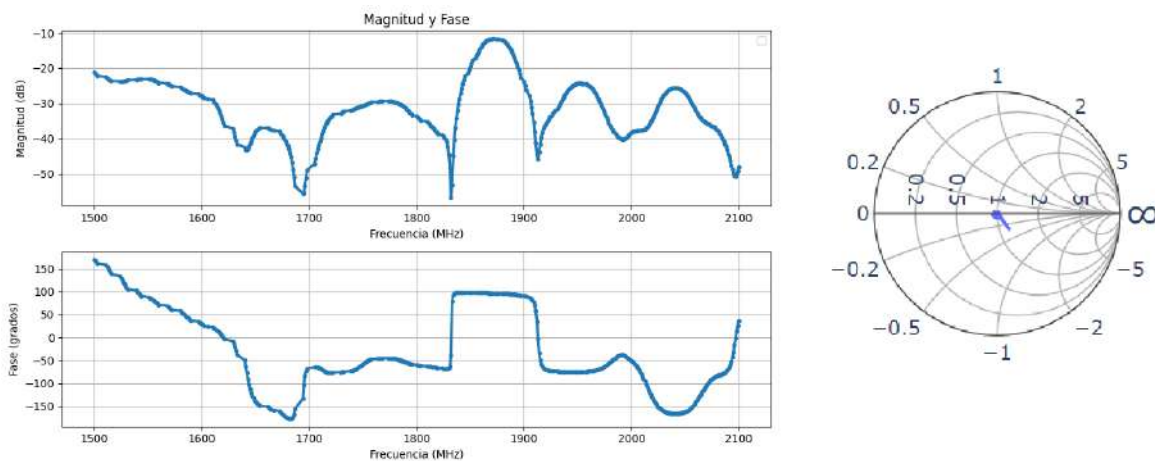


Figura 80: Medición de Carga 50 Ohm. Magnitud y fase.



## Carga Circuito Abierto

Se espera que esta medición sea 0 dB en magnitud y  $0^{\circ}$  en fase, con los puntos de medición en el extremo derecho del ábaco de Smith:

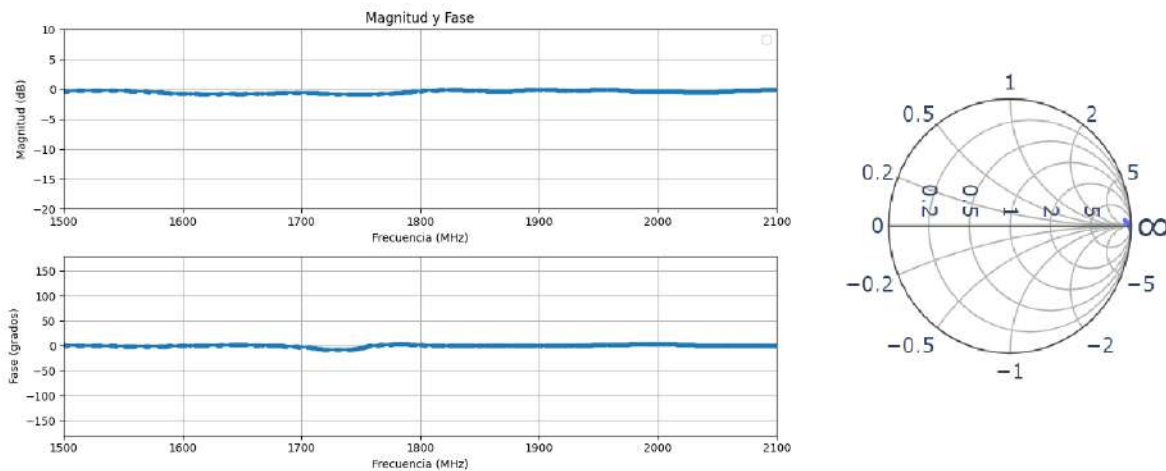


Figura 81: Medición de Carga Circuito Abierto. Magnitud y fase.

### 7.1.2. Transmisión

Para inicializar estas pruebas se conecta el USB CODEC S21/S12. Luego, se ajusta el barrido deseado dentro de UHF. Se emplea como salida el puerto ZL y el puerto de entrada TX. Para medición se emplean dos cable de RF entre ambos puertos unidos mediante un conector. Se adquieren las señales de transmisión según los siguientes pasos, para así obtener los coeficientes de error, generar la matriz de calibración y medir otra vez el mismo cable como un DUT deseado.

1. Conectar Puertos ZL y Tx (VNA) a carga  $50 \Omega$ . Presionar botón LEAK y medir con MEASURE. Generar vector de error  $e_{30}$ .
2. Interconectar Puertos ZL y Tx (VNA) con el cable con el que se interconectaría el DUT. Presionar botón THR y medir con MEASURE. Generar vector de error  $e_{1032}$ .
3. Conectar el DUT con el cable anterior. Presionar botón Sx y medir con MEASURE. Generar vector  $S_{21}$  o  $S_{12}$ .
4. Presionar CALIBRATE para calibrar. Luego, PLOT para graficar el parámetro medido y corregido.

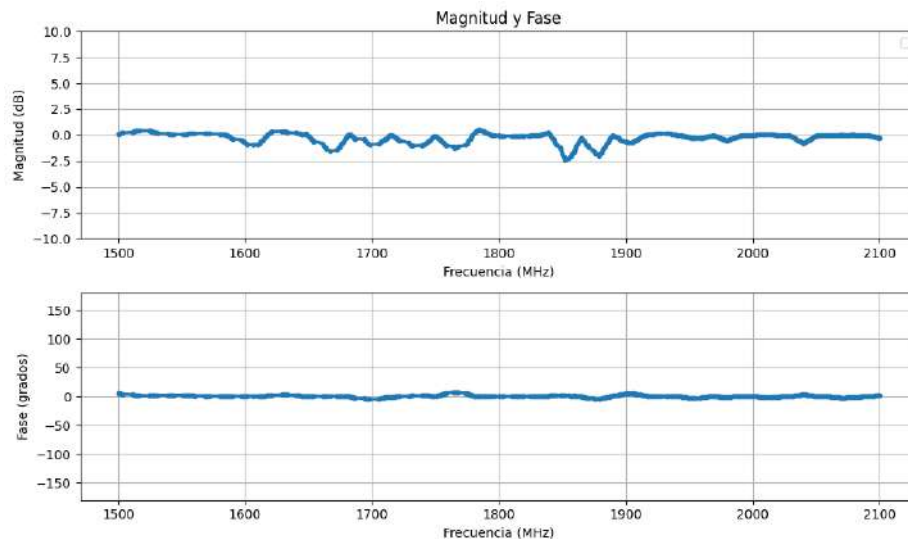


Figura 82: Medición de Cable THR. Magnitud y fase.

Como puede observarse en todas las figuras 79, 80, 81 y 82, correspondientes a las curvas de las mediciones de cargas y cables para la calibración, a pesar de que en general presentan la característica esperada, siguen habiendo errores a lo largo de toda la banda medida. Esto se debe tanto a las imperfecciones de las cargas como a la sumatoria de ruido aleatorio en la medición. Estos errores a su vez pueden afectar de manera significativa al momento de medir los parámetros S de un DUT cualquiera.

## 7.2. Mediciones DUT

A partir de las pruebas de calibración de la subsección anterior, se realizan mediciones de diferentes DUT para corroborar el funcionamiento del sistema. Para ello, se miden los DUT con el Analizador Rigol RSA3030N [2] para comparar los datos. Para mejorar levemente el suavizado de las señales y minimizar la influencia del ruido, se emplea una función de filtrado con el método Savitzky-Golay [37] (Anexo 9), el cual se basa en el cálculo de una regresión polinomial local (de grado  $k$ ), con al menos  $k+1$  puntos equiespaciados, para determinar el nuevo valor de cada punto. Es decir, que aplica un filtrado a partir de la quinta derivada de la función de puntos. A pesar de esto, todas las mediciones efectuadas son de un barrido único, es decir, no se aplica promediado de muestras.

En las siguientes gráficas se dibujan las mediciones realizadas de 3 DUT diferentes, en las cuales en cada una se diferencian: en color rojo, la curva de los datos extraídos de la medición del instrumento Rigol; en color rosado, se muestran los puntos de las mediciones de este proyecto VNA, y por último, en color azul, las curvas de suavizado.

### 7.2.1. Resonador 2 GHz

En la figura 83 se muestra la comparación de las mediciones de reflexión (Parámetro S11) de un sensor cuya frecuencia de resonancia de 20dB se espera que esté alrededor de 2 GHz. El barrido realizado es desde 1700 MHz hasta 2100 MHz, en pasos de 5 MHz.

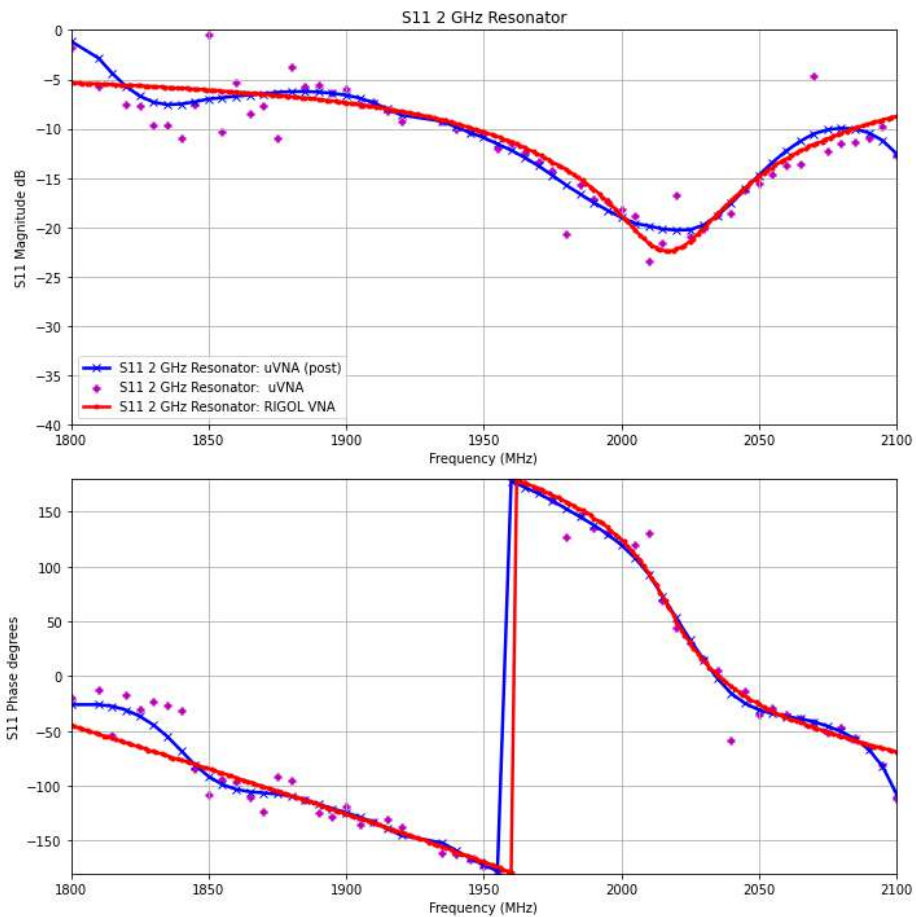


Figura 83: Medición de resonador 2 GHz. Parámetro S11.

Se observa que la frecuencia de resonancia es notable y se encuentra en aproximadamente 2010 MHz.

La figura 84 muestra el ábaco de Smith de las mediciones de la gráfica anterior:

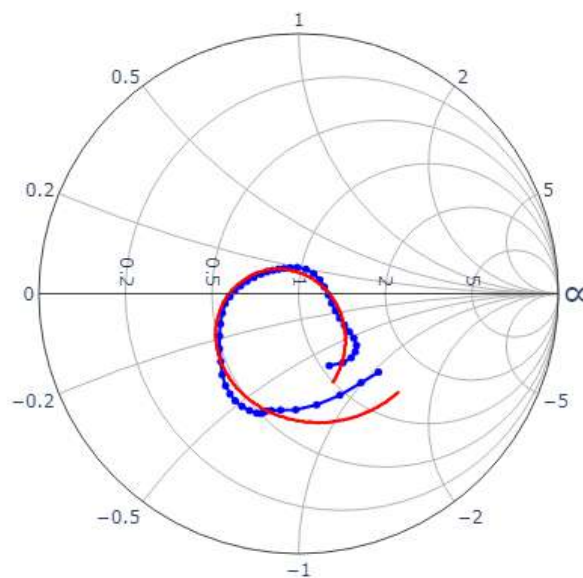


Figura 84: Medición de resonador 2 GHz. Ábaco de Smith.

En ambas gráficas se observa que el comportamiento de la curva obtenida a partir de los datos del dispositivo construido sigue la característica de la curva de medición del instrumento profesional. No obstante, se puede cuantizar un error en magnitud máximo de 5dB y un error de fase máximo de 25°

### 7.2.2. Atenuador 10dB

En la figura 85 se muestra la comparación de las mediciones de transmisión (Parámetro S21) de un atenuador de 10 dB. Se realiza un barrido desde 300 MHz hasta 2100 MHz, en pasos de 50 MHz.

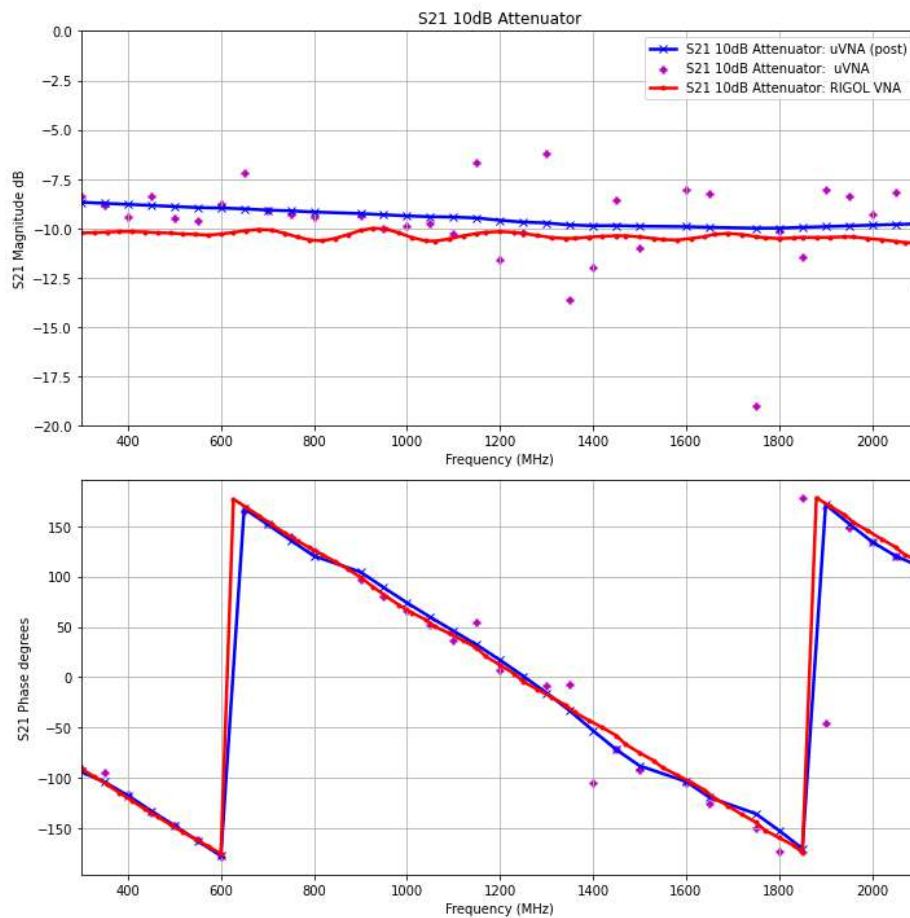


Figura 85: Medición de atenuador. Parámetro S21.

Si bien es notable el error en magnitud a lo largo de toda la banda, el cual es de 2 dB en el peor caso, la medición de fase posee buena precisión respecto a la medición hecha con el instrumento profesional.

### 7.2.3. Filtro 1,8 GHz

Se realiza la medición de un filtro de RF cuya frecuencia central es 1,8 GHz con un ancho de banda de 200 MHz. Se realiza un barrido desde 1600 MHz hasta 2000 MHz, en pasos de 10 MHz.

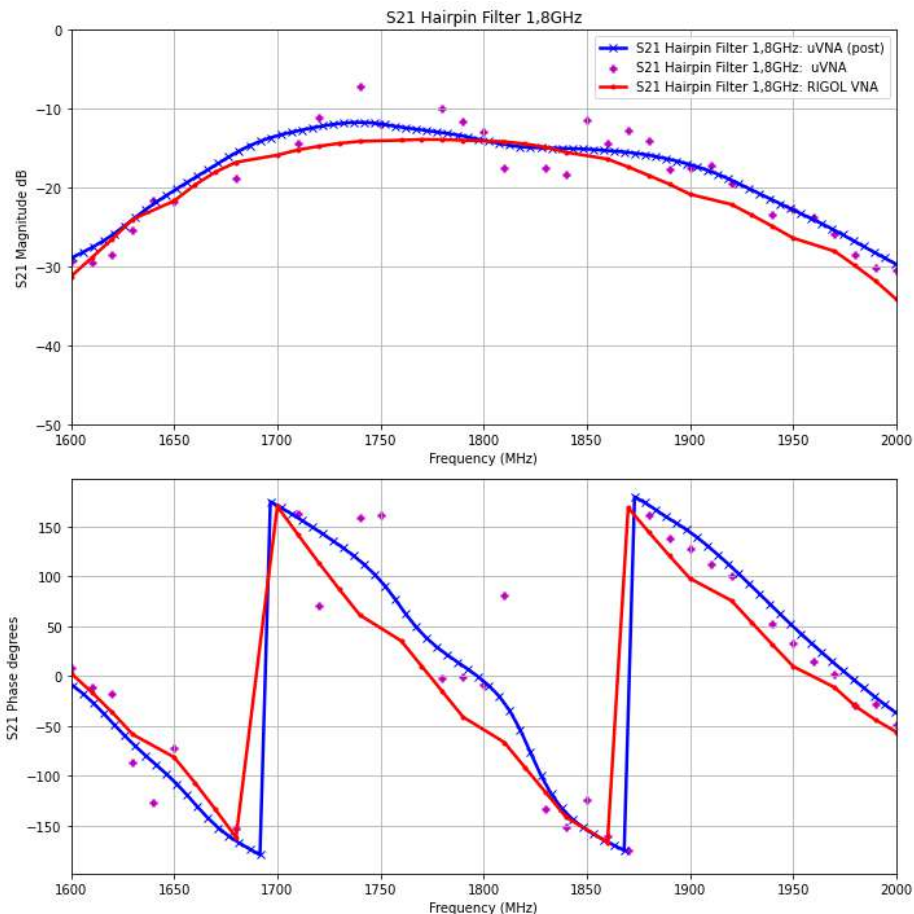


Figura 86: Medición de filtro 1,8 GHz. Parámetro S21.

Las oscilaciones que se presentan en la medición pueden haberse debido a un error de calibración. A pesar de esto, la curva obtenida tiende a seguir el comportamiento de los resultados obtenidos con el otro aparato. Se observa que en este caso la medición de fase posee mucho más error, en contraste con la medición del atenuador.

### 7.3. Interpretación de resultados

Desde el punto de vista del comportamiento, los resultados obtenidos en reflexión prueban el buen funcionamiento del acoplador resistivo direccional para obtener el parámetro de reflexión S11 o S22. Incluso puede brindarnos información certera acerca de la adaptación de impedancias de un sistema o bien de discontinuidades tales como corto circuitos o circuitos abiertos. Por otro lado, los resultados obtenidos en transmisión prueban el funcionamiento del sistema para obtener el parámetro de transmisión S21 o S12.

No obstante, desde la precisión, se identifica una limitación importante. El error observado en la comparación con las mediciones del instrumento profesional revelan que el sistema VNA tiene una gran dispersión al momento de obtener mediciones de reflexión y transmisión. Esto puede mejorarse aumentando la cantidad de muestras para promediar las señales o bien aplicando mejoras tanto en el procesamiento de señales como en la graficación de las mismas, ya sea con técnicas de filtrado o interpolación de datos.

## 8. Conclusiones y trabajo futuro

### 8.1. Logros principales

El objetivo principal del proyecto fue desarrollar un analizador vectorial de redes para la banda de frecuencias UHF, que sea capaz de medir parámetros S de diferentes sistemas bajo prueba. Se concluye entonces que el cometido se logró con éxito, pudiendo haber desarrollado un sistema tanto a nivel Hardware como a nivel Software, en el cual, a pesar de tratarse de un primer prototipo y de haber obtenido mediciones con errores considerables, los resultados obtenidos son comparables con los de un instrumento comercial similar.

Por otro lado, desde el enfoque de la línea de investigación de circuitos de microondas, se ha probado de manera experimental que el puente resistivo direccional funciona como un acoplador direccional.

Ahora bien, desde el punto de vista de las especificaciones:

- Obtención de parámetros S: Se han medido cargas de calibración (Corto circuito, circuito abierto y  $\Omega$ ), cables y conectores, atenuadores, resistores SMD, sensores de resonancia y filtros de RF.
- Configuración de medición y calibración: el usuario desde el Software del sistema puede configurar el rango deseado y poder realizar las iteraciones de calibración para las mediciones de reflexión y transmisión correspondientes.

y los requerimientos de rendimiento:

- Rango de medición de 1,8 a 2 GHz: El sistema funciona con mediciones aceptables desde 300 MHz hasta 2100 MHz.
- Impedancia característica de 50  $\Omega$ : ambos puertos de RF (ZL y TX) del sistema son de 50  $\Omega$ .

### 8.2. Recomendaciones y trabajo futuro

A pesar de cumplimentar los requisitos preestablecidos del proyecto, se brindan algunas recomendaciones para trabajo futuro.

#### Hardware

En primer lugar se aconseja utilizar otro tipo de material para el diseño de PCB. Esto debido a que las placas de FR4 poseen grandes pérdidas en UHF por lo que utilizar placas de mayor calidad podrían disminuir notablemente las pérdidas de las señales de RF.

Por otro lado, se recomienda implementar un oscilador de referencia más estable para los generadores ADF4351. Esto debido a que el error de fase del oscilador actual puede generar errores en la generación de las señales deseadas y el desenganche del circuito de PLL de cada generador.

Para el caso del acoplador direccional resistivo, una optimización en la posición de los resistores podría lograr una mejora en la respuesta en frecuencia de los parámetros de este arreglo.

Al mismo tiempo, el acoplamiento de cada línea de transmisión no ha sido corroborado en el diseño por lo que una minimización de esta característica podría mejorar las mediciones de reflexión.

La característica no lineal de los Mixers IAM81008 no ha sido tenido en cuenta en este proyecto por lo que se recomienda prestar atención en la utilización de este circuito integrado en cualquier tipo de proyecto.

La utilización de dos CODECs PCM2904 de audio se debió principalmente a la cantidad de líneas de señales de IF y a la característica de poseer el protocolo USB, el cual era beneficioso para el alcance del proyecto. No obstante, emplear un solo circuito integrado y un solo canal de comunicación, tanto en la recepción de señales como en el envío de frecuencias mediante el conversor USB/UART hacia el Microcontrolador PIC, lograría disminuir la cantidad de circuitería de la placa. A pesar de que el PIC 18F4550 utilizado posee la cantidad de canales necesaria, es cierto que necesitaría un tiempo de procesamiento mayor y se estaría desaprovechando los 16 bits de los ADC de los CODECs, en contraste con los 10 bits de los ADC del PIC. Una buena elección de un circuito integrado cuyas características de conversión analógica/digital y de comunicación digital sería de gran implicancia en el desarrollo de trabajos posteriores.

## Mediciones

Para mejorar la exactitud de las mediciones es necesario minimizar el tiempo de adquisición y procesamiento de señales a lo largo de todo el barrido. De esta manera, podrían tomarse una mayor cantidad de muestras por frecuencia, disminuyendo mediante la técnica de promediado a la dispersión de las mediciones, causadas por errores en la calibración, desperfecto de los conectores y cargas empleadas para esta última, o simplemente por adición de ruido aleatorio.

## Software

Si bien no era requisito para el proyecto, una integración entre el Backend principal del Software en Jupyter y el FrontEnd de la interfaz desarrollada es necesario para que el proyecto alcance su óptimo funcionamiento desde la interacción y experiencia del usuario. Por otro lado, si se desea implementar el proyecto en un instrumento completamente portátil, podría embeberse todo el sistema en un microcontrolador en conjunto con una pantalla táctil y botones. De esta manera, se estaría independizando de su instalación en una computadora personal.

## 9. Anexos

### 9.1. Diseño de PCB en RF

#### 9.1.1. Pérdidas y efectos indeseados

En el diseño de sistemas circuitos electrónicos existen diversos problemas que hay que tener en cuenta a la hora de realizar la implementación real en una placa PCB [38]. Entre ellos se encuentran:

- Pérdidas debidas al material FR4: tangente de pérdidas, variación de la constante dieléctrica, etc.
- Diferencias de fase ocasionadas por caminos de gran longitud respecto a  $\lambda_{ef}$ .
- Efecto pelicular o skin.
- Interferencias y/o autoresonancias.
- Corrientes de retorno parásitas.

Las mayores pérdidas y efectos parásitos que pueden afectar al funcionamiento de un circuito de RF en general se deben al material FR4 que se utiliza para construir el PCB. Este material tiende a atenuar las señales mayores a 2GHz por lo que no es recomendable utilizarlo en frecuencias mayores. No obstante, se emplea debido a su bajo costo y su sencilla adquisición. La tangente de pérdidas, por ejemplo, se define como una medida de la disipación de energía:

$$\tan \delta = \frac{\epsilon''}{\epsilon'} \quad (46)$$

Donde  $\epsilon''$  es la parte imaginaria de la permitividad compleja del material dieléctrico, mientras que  $\epsilon'$  es su parte real. En la siguiente gráfica se observa las pérdidas en función de la frecuencia de un material FR4 vs un material tipo Rogers de mayor calidad:

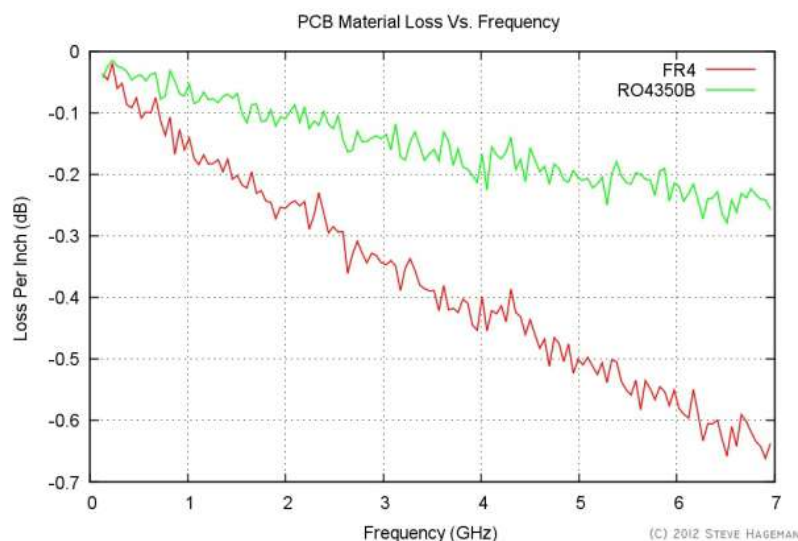


Figura 87: Pérdidas por pulgada en función de la frecuencia: FR4 vs Rogers. Fuente: [39].

Por otro lado, los demás problemas pueden mitigarse o solucionarse si se siguen ciertas reglas de diseño de circuitos de microondas.



### 9.1.2. Compatibilidad Electromagnética

Para satisfacer la compatibilidad electromagnética de un sistema de RF, toda señal de microondas tiene que tener su corriente de retorno y tal corriente debe volver por el camino de menor inductancia, que suele ser en el plano inferior al trazo de la línea de transmisión para una placa PCB [13]. El concepto de GND de menor potencial se expande aquí ya que existen diversos tipos de GND en un aparato y debe tenerse muy en cuenta que toda señal tiene que tener un camino liberado de baja inductancia para su corriente de retorno. En placas PCB, normalmente este camino es inmediatamente debajo de la línea de transmisión por el cual se propaga, el cual tiene sentido debido a las líneas de campo eléctrico:

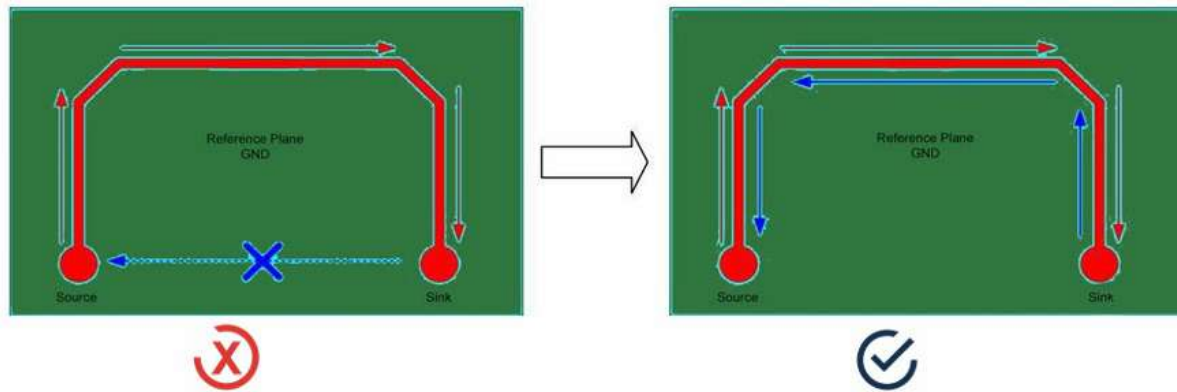


Figura 88: RF Return Path. Fuente: [13].

Ahora bien, hay que tener presente que a pesar de que las líneas de transmisión en un circuito PCB están rodeadas de secciones sin conductor eléctrico, la realidad es que las señales de RF encuentran camino por todos lados, es decir, que se propagan por cualquier medio material con o sin atenuación. Esto significa que hay que lograr controlar no solamente las señales indeseadas interferentes si no también controlar las señales deseadas. De esto trata la compatibilidad electromagnética.

Si se tiene una sección de cobre de tamaño considerable en términos de longitud de onda en una placa PCB, esta actuará como una antena que capte toda onda electromagnética que induzca corriente en dicha superficie, generando que se produzca una autoresonancia en dicha sección y que a su vez se propague por el resto de la placa, generando interferencia. La expresión siguiente indica la frecuencia de la autoresonancia para una superficie de tamaño  $axb$ :

$$f_o = \frac{c_o}{2\sqrt{\epsilon_r} \cdot \sqrt{a^2 + b^2}} \quad (47)$$

Para solventar este problema, por ejemplo, es tan sencillo de colocar vías entre la sección para cortar el camino de la interferencia y elevar en frecuencia a estas autoresonancias, que se verán mayoritariamente atenuadas debidas al material del PCB. Esta técnica se denomina Vía Stitching [18]. Esta técnica también se emplea para mejorar el camino de las señales de retorno debido a que múltiples vías en paralelo poseen una inductancia muy pequeña entre el plano superior y el plano inferior. Por otro lado, también sirve para que haya mayor área de dispersión térmica en dichos planos.

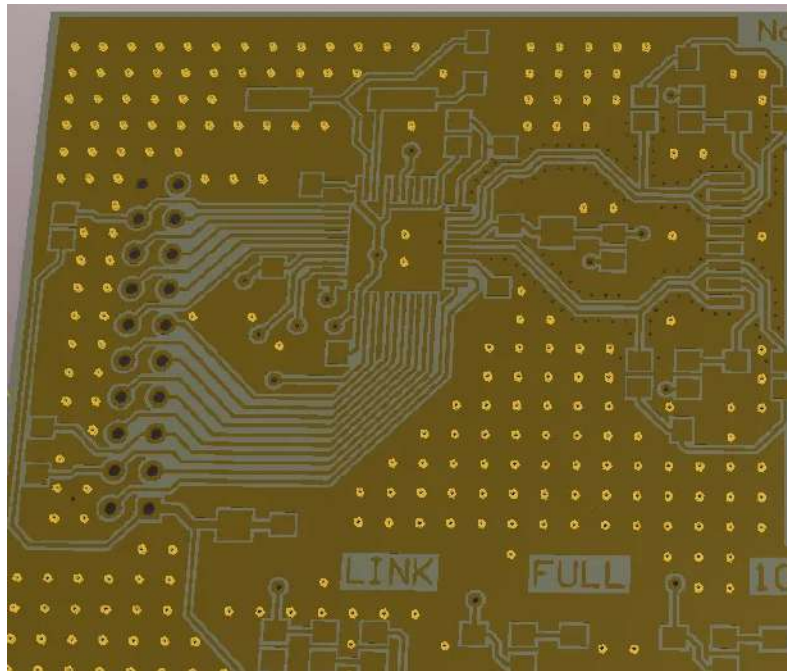


Figura 89: Técnica Vía Stitching: Top GND - Bottom GND. Fuente: [18].

En líneas de transmisión existen diversas técnicas para producir la propagación de una señal mediante un conductor en un PCB. Entre las más conocidas existen las líneas basadas en microstrip [4]:

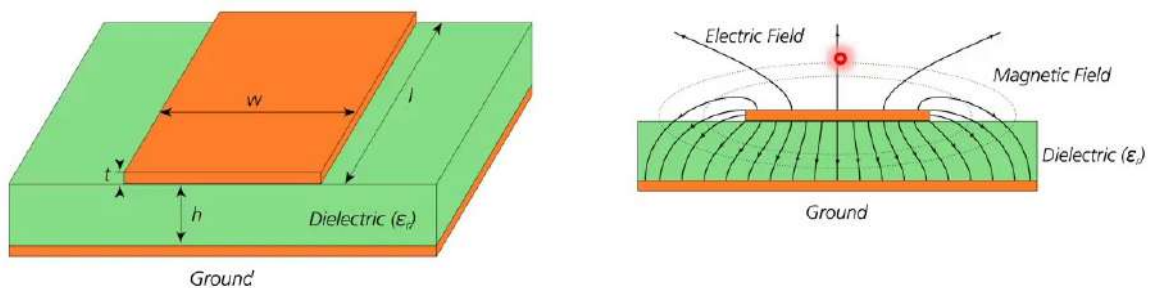


Figura 90: Técnica Microstrip. Fuente: [4].

No obstante, para disminuir acoplamientos (Crosstalk [13]) y aumentar el aislamiento a interferencias, existe la técnica de líneas coplanares [4], que vendría a ser una línea microstrip pero rodeada de planos de GND para cortar las líneas de campo eléctrico.

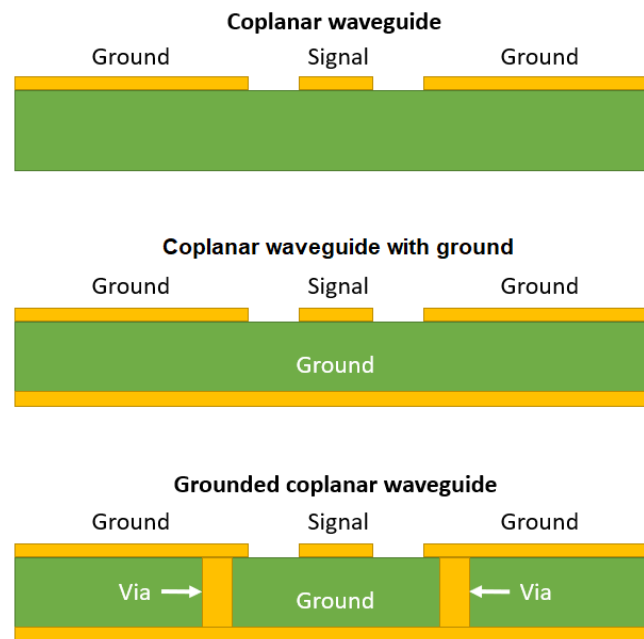


Figura 91: Técnica Líneas Coplanares. Fuente: [4].

Para aumentar la performance del aislamiento se emplean vías a GND con una regla basada en la longitud de onda mínima del sistema para la distancia entre vías. Esta técnica se denomina *via fencing* o *via shielding* [18].

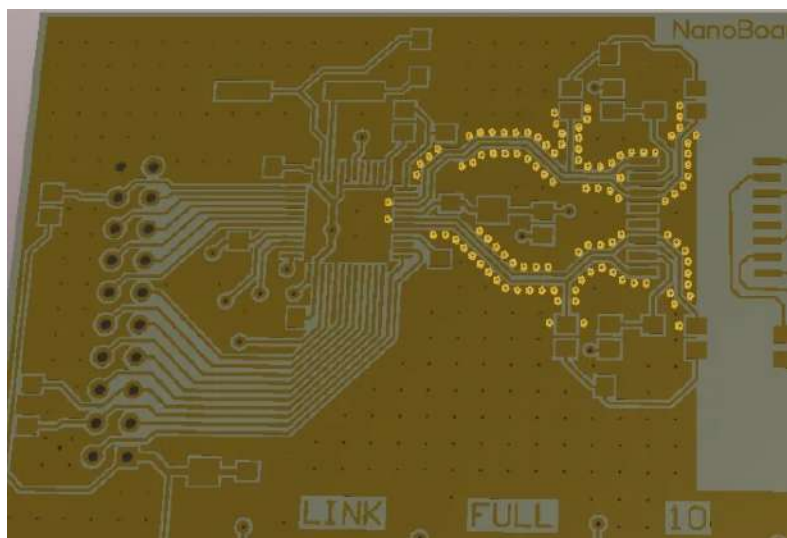


Figura 92: Técnica *via fencing*: Top GND - Bottom GND. Fuente: [18].

En resumen, antes de realizar un análisis para mejorar la integración de la señal (*Signal integration*) o análisis de las peores situaciones (*Worst cases*), si se trabajan con señales de microondas o incluso de alta velocidad en circuitos digitales (donde las componentes armónicas son de gran importancia), hay que tener en cuenta el concepto de compatibilidad electromagnética y tratar de seguir todos los lineamientos de diseño para así disminuir los efectos indeseados. Claramente, al tratarse de circuitos de RF, siempre van a haber problemas pero queda en el diseñador encontrar las mejores prácticas para mejorar el funcionamiento o *performance* del sistema en cuestión.

## 9.2. Acoplador resistivo direccional

### Resumen

Esta sección se enfoca en el diseño de un acoplador bidireccional para la banda de frecuencias UHF basado en un puente de Wheastone [19] e implementado con resistores discretos de tecnología SMT, cuyas características en frecuencia permiten obtener resultados de gran ancho de banda. El arreglo permite tomar una muestra de la señal incidente que se inyecta sobre la carga incógnita y es capaz de obtener una señal que representa a la señal reflejada.

El objetivo de este acoplador es que sea utilizado en el analizador vectorial de redes, también desarrollado por el grupo, para poder caracterizar equipos de radiofrecuencia (o simplemente dispositivos bajo prueba o DUT) en términos de parámetros S, dentro de la banda estipulada.

### Vector Network Analyzer Architecture

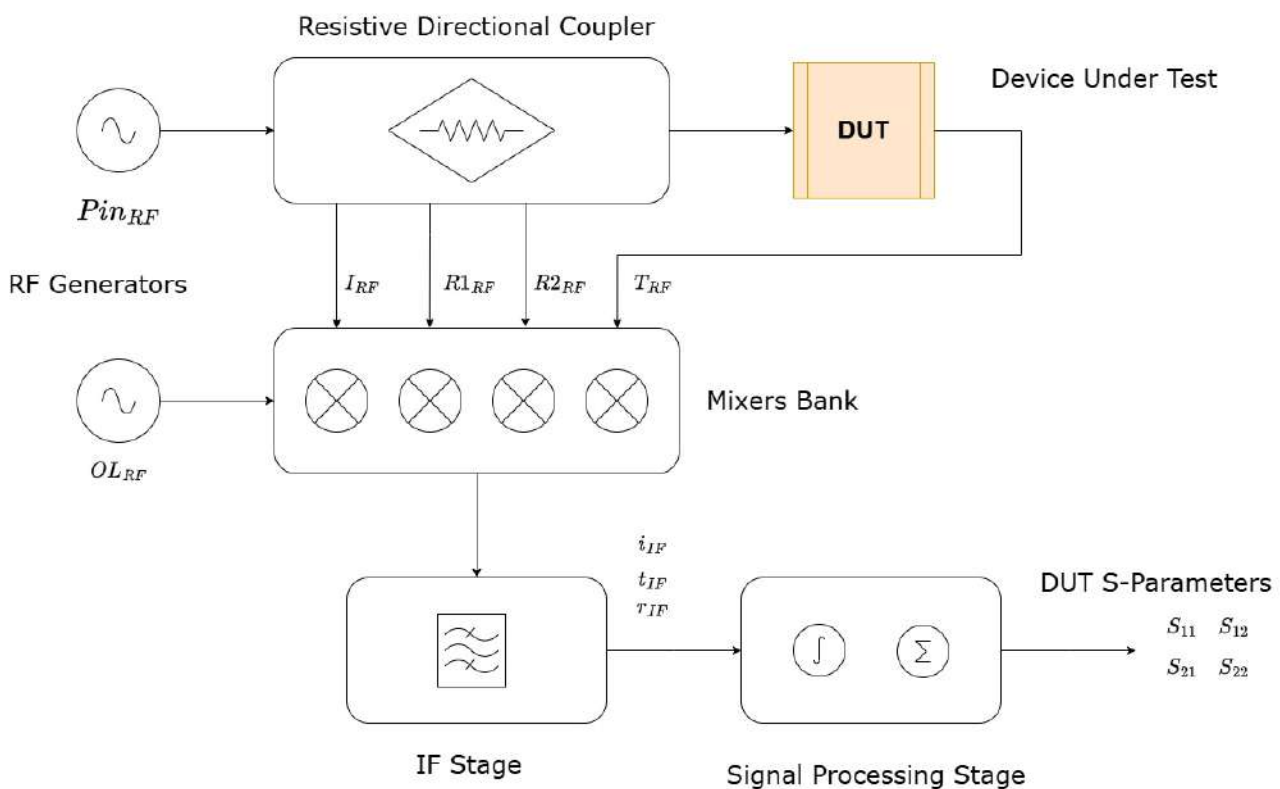


Figura 93: Arquitectura del analizador vectorial de redes propuesto.

### Especificaciones

Las especificaciones que debe cumplir el acoplador direccional resistivo son:

- Obtener una muestra de la señal incidente a través del puerto 3 y una muestra de la señal reflejada en la banda de UHF en el puerto 4.
- Obtener la mínima pérdida de inserción y la máxima directividad, si es posible.
- Todos los puertos deben trabajar con impedancia de  $50 \Omega$ .

- Se utilizarán como DUT a cargas corto circuito (CC), circuito abierto (CA) y  $50 \Omega$
- Adaptar las salidas de los puertos de acoplamiento a una etapa de banco de Mixers para su posterior procesamiento en baja frecuencia.

### 9.2.1. Marco Teórico

En la figura 94 se observa el acoplador direccional propuesto en términos de parámetros concentrados.

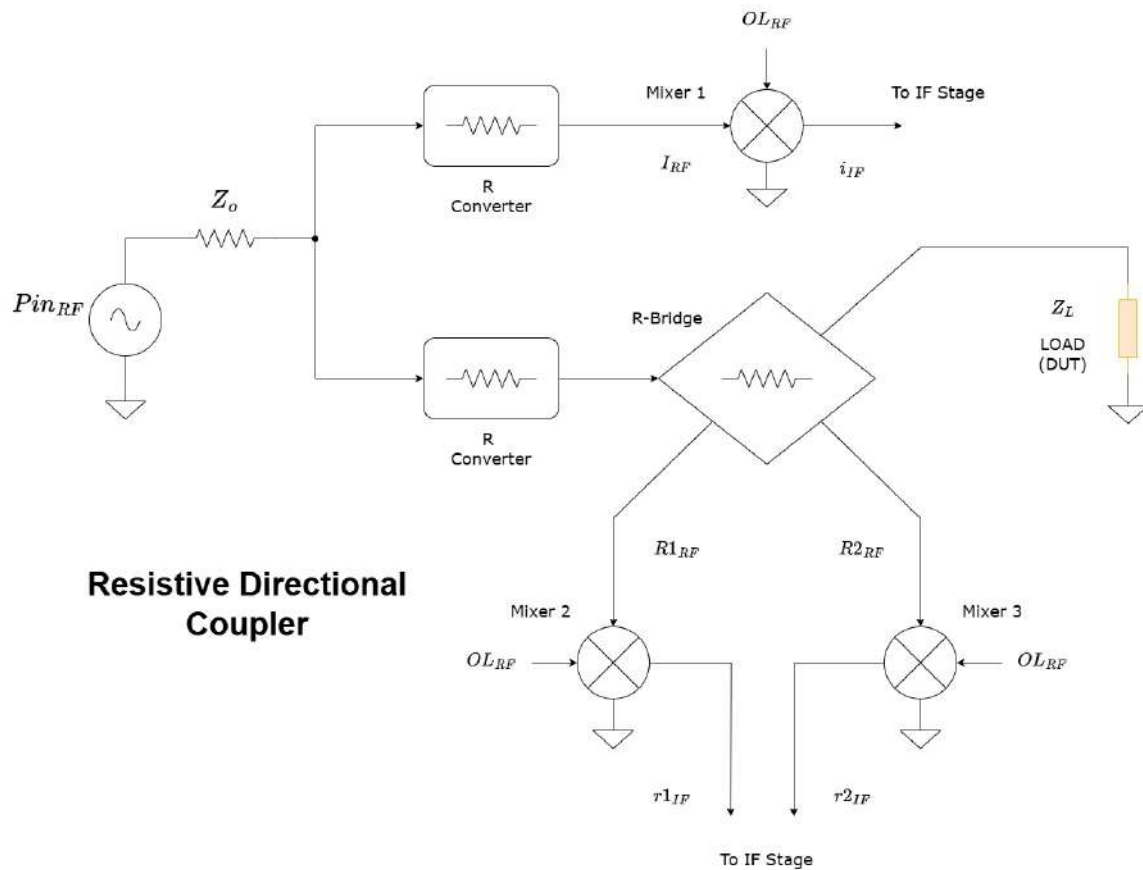


Figura 94: Acoplador direccional propuesto.

Teniendo como fuente de excitación a un generador de RF barredor de  $50\Omega$ , se emplea un divisor de potencia resistivo [4] para bifurcar el camino de la señal mediante dos ramas. La primera rama se emplea para tomar una muestra de la referencia o señal incidente hacia la próxima etapa mediante el primer Mixer. La segunda rama es la que se utiliza para el puente resistivo, que transfiere la potencia hacia el puerto de carga y entrega dos señales cuya diferencia representa a la señal reflejada, que van hacia la próxima etapa. En ambas ramas se emplean adaptadores de impedancia para minimizar las pérdidas de inserción.

#### Divisor de potencia

El divisor de potencia [4] que se utiliza es bastante elemental y presupone las características de impedancia en términos de parámetros distribuidos. Si la impedancia de salida del generador es de  $50 \Omega$ , simplemente la impedancia vista por cada rama debe ser de  $100 \Omega$ , donde la rama A es el camino de la muestra de la señal de referencia y la rama B es el camino hacia el puente resistivo y la carga.

De esta manera, la potencia se distribuye de manera uniforme tal que la potencia en cada rama es:

$$P_A = \frac{P_{in}}{4}(-6dB) \quad (48)$$

$$P_B = \frac{P_{in}}{4}(-6dB) \quad (49)$$

### Rama A: señal de referencia

Algunos circuitos multiplicadores integrados en chips MMIC [22] (cuya finalidad en este caso es bajar en frecuencia a las señales para que sean más sencillas de procesar) presentan en general una impedancia de  $50\ \Omega$  en sus terminales de entrada de RF. Por esta razón se agrega un adaptador de impedancia con una red resistiva de parámetros concentrados tal como se observa en la figura 95.

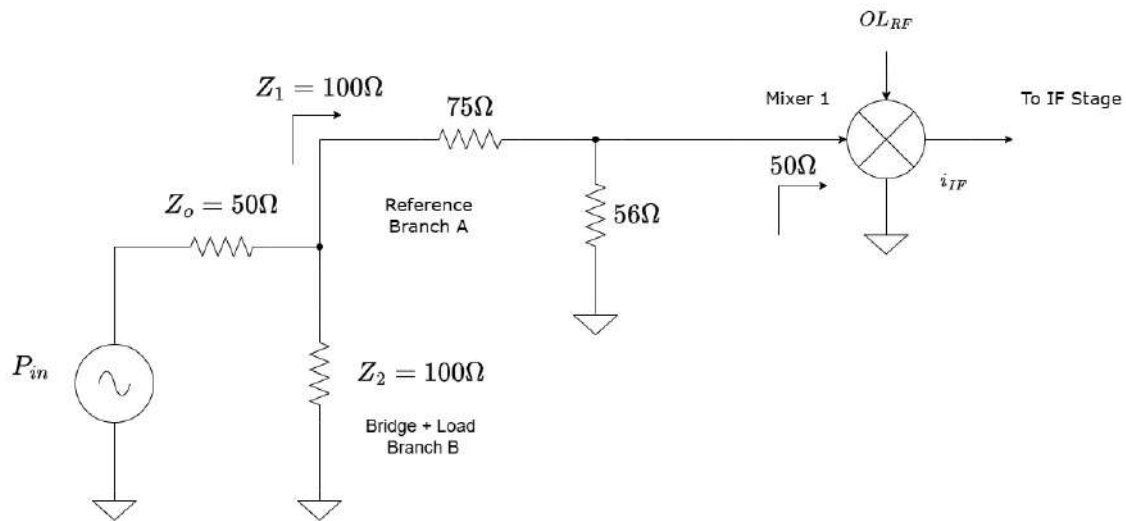


Figura 95: Rama A de referencia.

La potencia que recibe el primer Mixer es de aproximadamente:

$$P_3 = \frac{P_{in}}{4 \cdot 4}(-12dB) \quad (50)$$

En términos de parámetros S, el acoplamiento que se obtiene es de:

$$C = 10 \cdot \log \frac{P_{in}}{P_3} = 12dB \quad (51)$$

$$S_{31} = 0.25 \quad (52)$$

De esta manera, la impedancia que presenta la rama A al divisor de potencia es de  $100\ \Omega$ .

### Rama B: Puente resistivo ideal

El principio de funcionamiento del puente resistivo está basado en el puente de Wheastone [19], cuya función es medir la impedancia incógnita de la carga a partir de la medición de la diferencia de tensión en el arreglo:

$$V_G = I_G \cdot R_G = V_x - V_2 \tag{53}$$

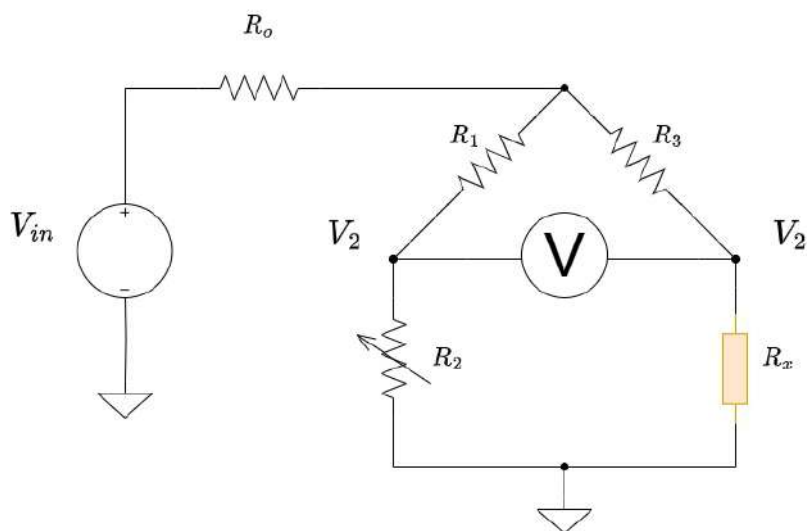


Figura 96: Puente de Wheastone.

A partir de la segunda Ley de Kirchoff:

$$\begin{cases} I_3 R_3 - I_G R_G - I_1 R_1 = 0 \\ I_x R_x - I_2 R_2 + I_G R_G = 0 \end{cases} \tag{54}$$

Si el puente está balanceado tal que  $V_G = 0$ , la resistencia incógnita se puede calcular como:

$$R_x = \frac{R_2 \cdot R_3}{R_1} \tag{55}$$

A partir de este principio, se presenta el puente resistivo balanceado a una impedancia de  $50 \Omega$ , tal que  $R_1 = R_2 = R_3 = R_G = R_s = 50 \Omega$ :

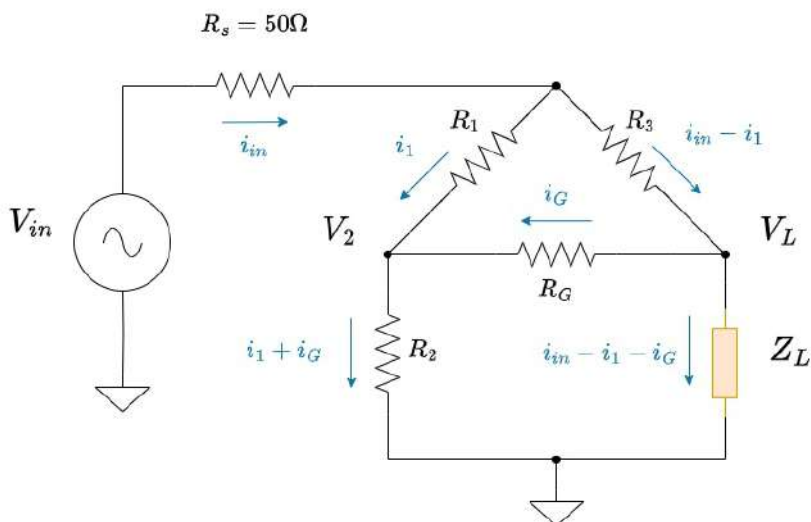


Figura 97: Puente propuesto balanceado.

Esto implica por simple observación que cuando  $V_g = V_L - V_2 = 0$ , entonces la carga es  $Z_L = 50 \Omega$ .

A partir de la primera y segunda Ley de Kirchoff:

$$\begin{cases} v_{in} = i_{in}R_s + i_1R_s + (i_1 + i_G) \cdot R_s & \text{(a)} \\ v_{in} = i_{in}R_s + i_1R_s - i_GR_s + (i_{in} - i_1 - i_G) \cdot Z_L & \text{(b)} \\ v_{in} = i_{in}R_s + (i_{in} - i_1) \cdot R_s + (i_{in} - i_1 - i_G) \cdot Z_L & \text{(c)} \end{cases} \quad (56)$$

Comparando las expresiones 56-a y 56-b:

$$i_G \cdot (2R_s + Z_L) = -i_1 \cdot (R_s + Z_L) + i_{in}Z_L \quad (57)$$

Lo mismo para 56-b y 56-c:

$$i_1 = \frac{1}{2} \cdot (i_G + i_{in}) \quad (58)$$

Reemplazando la ecuación 58 en 57 y en 56-a:

$$i_G \cdot (5R_s + 3Z_L) = i_{in} \cdot (Z_L - R_s) \quad (59)$$

$$i_{in} = \frac{V_{in}}{2R_s} - i_G \quad (60)$$

Finalmente se reemplaza la ecuación 60 en 59, y trabajando algebraicamente se obtiene la tensión en el resistor  $R_G$ :

$$V_G = I_G R_s = -\frac{V_{in}}{8} \cdot \frac{R_s - Z_L}{R_s + Z_L} \quad (61)$$

Y dado que  $Z_o = R_s$ , el coeficiente de reflexión se determina como:

$$\Gamma = \frac{Z_o - Z_L}{Z_o + Z_L} \quad (62)$$

Por lo que la tensión en el resistor es proporcional al coeficiente de reflexión debido a la carga, multiplicado por la señal incidente, es decir, proporcional a la señal reflejada  $V_{rev}$ :

$$V_G = I_G R_s = -\frac{V_{in}}{8} \cdot \Gamma = C \cdot V_{rev} \quad (63)$$

No obstante, esto sería posible de medir si tan solo se pudiera obtener una muestra de las tensiones de los terminales del resistor  $R_G$ , es decir,  $V_L$  y  $V_2$ . Esto podría realizarse si se utilizase un voltímetro ideal cuya impedancia sea lo suficientemente grande para no cargar al puente resistivo.

### Rama B: Puente resistivo realizable

Para que el puente resistivo sea realizable, se emplean como puntas de voltímetro a los Mixers de  $50 \Omega$  para luego bajar en frecuencia a las señales y poder obtener de manera más sencilla la diferencia  $V_G$  de las tensiones  $V_L$  y  $V_4$ .

Las ramas de cada medición deben presentar al menos 10 veces la impedancia  $R_s$ . Este valor se adopta debido al compromiso entre no cargar demasiado al puente resistivo pero teniendo la suficiente resolución en términos de potencia para al menos discernir y adquirir cada señal. Se adopta la resistencia  $R_G$  como parámetro de diseño y ajuste tal que  $R_G = K \cdot R_s$ . El circuito modificado es el siguiente:



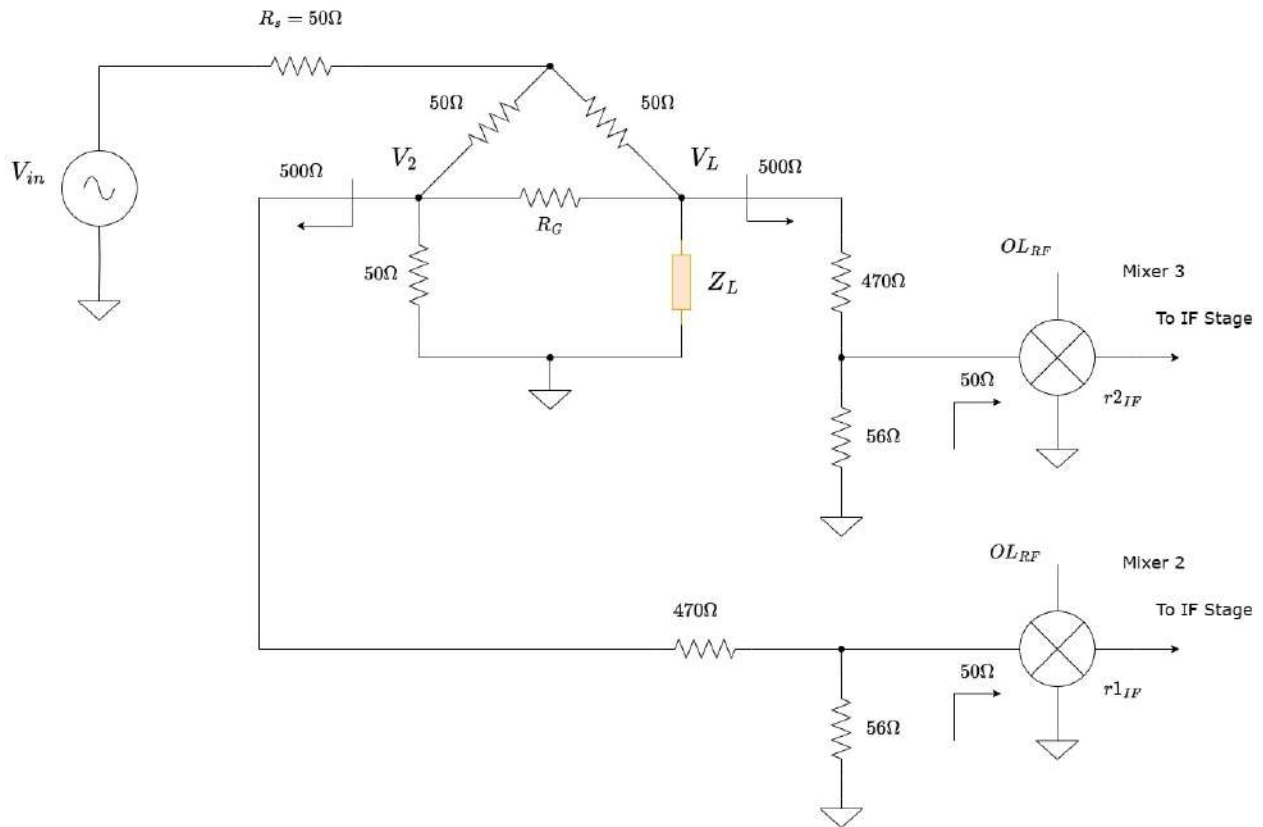


Figura 98: Puente propuesto cargado por la etapa de Mixers.

El circuito del puente resistivo simplificado en términos del puente resistivo ideal queda como:

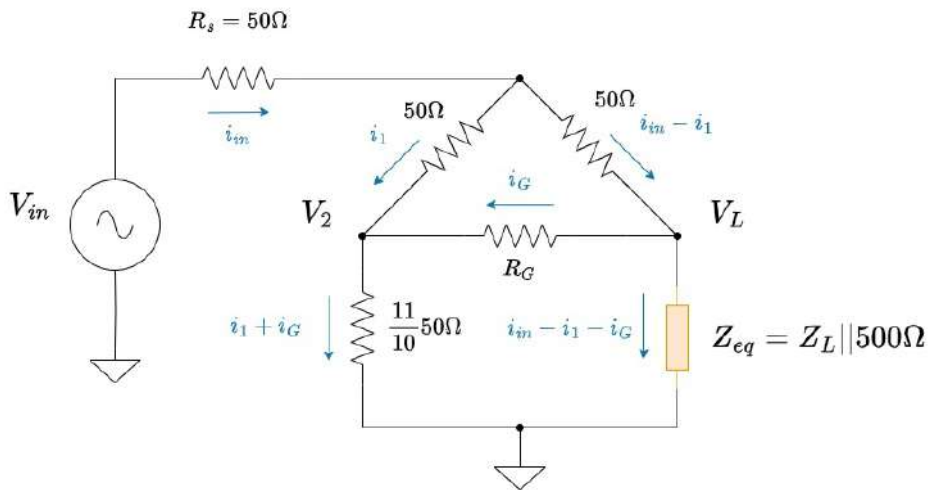


Figura 99: Puente propuesto cargado por la etapa de Mixers (simplificado).

Donde:

$$Z_{eq} = Z_L || 10R_s = \frac{Z_L \cdot 10R_s}{Z_L + 10R_s} \tag{64}$$

Repetiendo el análisis anterior:

$$\begin{cases} v_{in} = i_{in}R_s + i_1R_s + (i_1 + i_G) \cdot \frac{10}{11}R_s & (a) \\ v_{in} = i_{in}R_s + i_1R_s - i_GKR_s + (i_{in} - i_1 - i_G) \cdot Z_{eq} & (b) \\ v_{in} = i_{in}R_s + (i_{in} - i_1) \cdot R_s + (i_{in} - i_1 - i_G) \cdot Z_{eq} & (c) \end{cases} \quad (65)$$

Comparando las expresiones 65-a y 65-b:

$$i_G \cdot \left( \frac{10}{11}R_s + KR_s + Z_{eq} \right) = -i_1 \cdot \left( \frac{10}{11}R_s + Z_{eq} \right) + i_{in}Z_{eq} \quad (66)$$

Lo mismo para 65-b y 65-c:

$$i_1 = \frac{1}{2} \cdot (Ki_G + i_{in}) \quad (67)$$

Reemplazando la ecuación 67 en 66 y en 65-a:

$$i_G \cdot \left( \frac{10}{11}R_s + KR_s + \frac{K}{2} \frac{10}{11}R_s + Z_{eq} + \frac{K}{2}Z_{eq} \right) = \frac{1}{2}i_{in} \cdot \left( Z_{eq} - \frac{10}{11}R_s \right) \quad (68)$$

$$i_{in} = \frac{22}{43} \frac{V_{in}}{R_s} - \frac{22}{43} i_G \cdot \left( \frac{10}{11} + \frac{21K}{22} \right) \quad (69)$$

Finalmente se reemplaza la ecuación 69 en 68, y trabajando algebraicamente se obtiene la tensión en el resistor  $R_G$ :

$$V_G = I_G R_s = -\frac{22V_{in}}{2} \cdot \frac{\frac{10}{11}R_s - Z_{eq}}{R_s \cdot (30 + 53K) + Z_{eq} \cdot (53 + 32K)} \quad (70)$$

Reemplazando  $Z_{eq}$  en función de  $Z_L$  y  $R_s$ :

$$V_G = I_G R_s = -100V_{in} \cdot \frac{R_s - Z_L}{R_s \cdot (300 + 530K) + Z_L \cdot (560 + 373K)} \quad (71)$$

La ecuación de diseño es:

$$300 + 530K = 560 + 373K \quad (72)$$

Despejando se obtiene:  $K = 1,65605$ , lo que propone que  $R_G = 82,8 \Omega$ . De esta manera, la tensión sobre  $R_G$  es:

$$V_G = -0,0849V_{in} \cdot \frac{R_s - Z_L}{R_s + Z_L} \quad (73)$$

En términos del coeficiente de reflexión:

$$V_G = I_G R_s = -0,0849V_{in} \cdot \Gamma = C \cdot V_{rev} \quad (74)$$

Este resultado nos indica que, midiendo con los Mixers a las señales  $V_L$  y  $V_2$  y realizando su diferencia obtenemos una señal proporcional a la señal reflejada por la carga.

La relación entre las tensiones de entrada de los Mixers 2 y 3 y las tensiones de los terminales de  $R_G$  la podemos aproximar al divisor resistivo de cada rama:

$$V_{in}(MX2) = \frac{50 + 56}{50 + 56 + 470} \cdot V_2 = 0,182 \cdot V_2 \quad (75)$$

$$V_{in}(MX3) = 0,182 \cdot V_L \quad (76)$$

Ahora bien, conociendo las tensiones de entrada de los Mixers, se relaciona fácilmente el coeficiente de reflexión:

$$V_G = \frac{1}{0,182} \cdot (V_{in}(MX3) - V_{in}(MX2)) = -0,0849 \cdot V_{in} \cdot \Gamma = C \cdot V_{rev} \quad (77)$$

La diferencia entre estas tensiones es:

$$V_{in}(MX2) - V_{in}(MX3) = 0,015 \cdot V_{in} \cdot \Gamma \quad (78)$$

Por otro lado, lo que falta es adaptar la entrada del puente resistivo con la rama B de  $100 \Omega$ . Para ello hay que analizar cómo varía la impedancia que presenta el puente respecto a diferentes situaciones de carga. Por simple inspección del circuito de la figura 99 se puede calcular que la impedancia de entrada varía con la carga entre aproximadamente  $30,65 \Omega$  (corto circuito) a  $73,9 \Omega$  (circuito abierto). En el caso de adaptación con  $Z_L = 50 \Omega$ , la impedancia que se presenta es de  $47,72 \Omega$ , que, a fines prácticos es bastante aproximado a  $50 \Omega$ .

Para mantener la adaptación, se emplea la siguiente red resistiva tipo T, de atenuación 6dB:

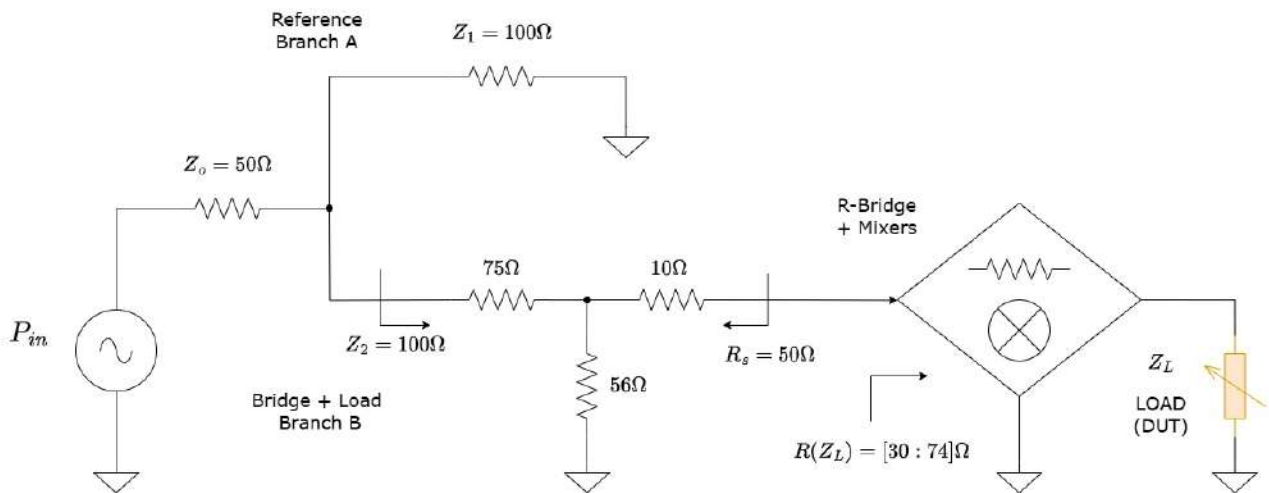


Figura 100: Agregado de red adaptadora.

Este arreglo de adaptación logra disminuir el efecto de la carga sobre la impedancia de la segunda rama (entre  $98 \Omega$  a  $108 \Omega$ ), logrando disminuir la reflexión hacia el generador y maximizando la transferencia de potencia desde dicho generador a la carga.

### 9.2.2. Diagrama final

Finalmente, el acoplador direccional basado en el puente resistivo para su implementación en un analizador vectorial de redes es:

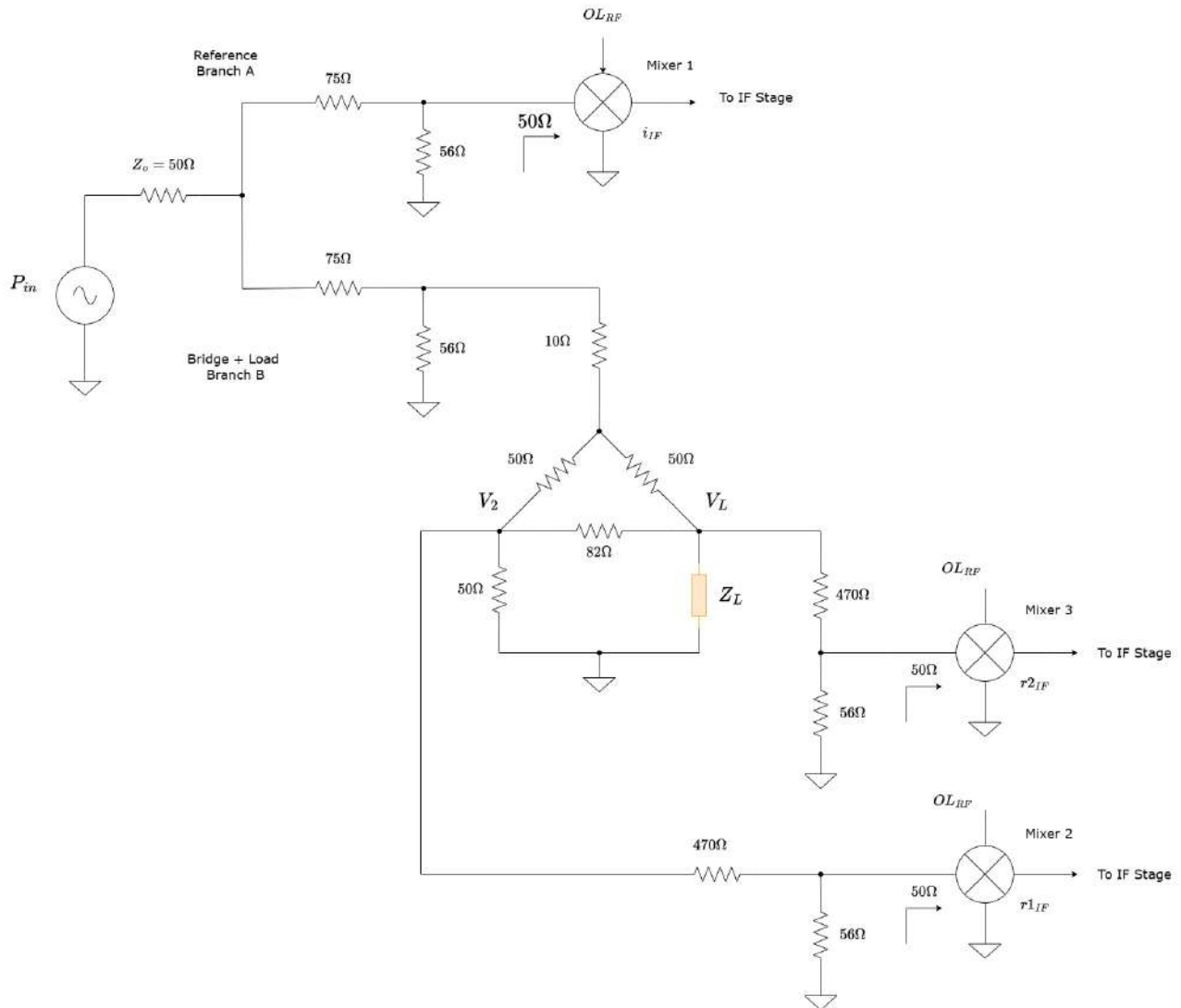


Figura 101: Red del acoplador direccional resistivo propuesto.

Desde el punto de vista de Parámetros S, el acoplador propuesto queda con 5 puertos:

- Puerto 1: Puerto incidente.
- Puerto 2: Puerto pasante (hacia carga del DUT).
- Puerto 3: Puerto acoplado al puerto 1 (señal de referencia o muestra de incidente).
- Puerto 4a: Puerto acoplado al puerto 2 (medición de variación en carga referencia de  $50 \Omega$ ).
- Puerto 4b: Puerto acoplado al puerto 2 (medición de variación de carga del DUT).

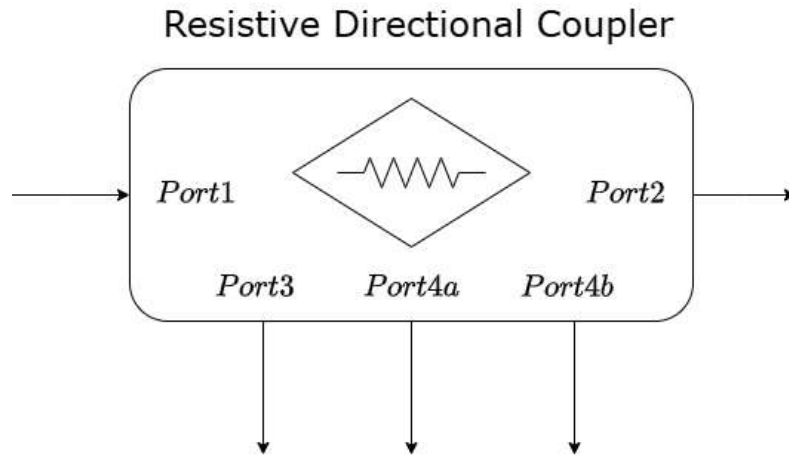


Figura 102: Diagrama en bloque del acoplador direccional resistivo propuesto.

La relación entre el coeficiente de reflexión y el parámetro  $S_{ii}$  correspondiente al puerto  $i$  de un sistema DUT de 2 puertos es:

$$S_{ii}(DUT)[dB] = 20 \cdot \log(|\Gamma_i|) \quad (79)$$

Por lo que el coeficiente de reflexión de entrada del DUT es simplemente:

$$S_{11}(DUT)[dB] = 20 \cdot \log(|\Gamma_i|) \quad (80)$$

A partir de la ecuación 35 se puede obtener fácilmente la relación entre los parámetros de salida del acoplador y el coeficiente de relación deseado:

$$V_{in}(MX2) - V_{in}(MX3) = 0,0139 \cdot V_{in} \cdot \Gamma \quad (81)$$

No obstante, hay que tener en cuenta que  $V_{in}$  es la tensión de entrada al puente no es la entrada al acoplador. La relación entre estas se debe al divisor de potencia y al adaptador de impedancias dando como relación:

$$\frac{V_{input}}{4} = V_{in} \quad (82)$$

Luego:

$$\frac{V_{in}(MX2)}{V_{input}} - \frac{V_{in}(MX3)}{V_{input}} = \frac{0,015}{4} \cdot \Gamma \quad (83)$$

$$S_{4a} - S_{4b} = \frac{0,015}{4} \cdot \Gamma \quad (84)$$

Por lo tanto el coeficiente de reflexión del DUT se puede calcular como:

$$\Gamma = 267 \cdot (S_{4a} - S_{4b}) \quad (85)$$

### Simulaciones

Para agilizar el proceso de cálculo, se emplea el software de simulación QUCS [10]. En la figura siguiente se representa al acoplador propuesto en la interfaz del programa. Se define al Puerto 1 como la entrada del generador de potencia, el Puerto 2 como la entrada del Mixer 2 que mide la tensión  $V_2$ , el Puerto 3 como la entrada del Mixer 3 que mide la tensión  $V_L$ , el Puerto 4 es el Mixer 1 de referencia y el Puerto 5 es la carga variable del DUT.

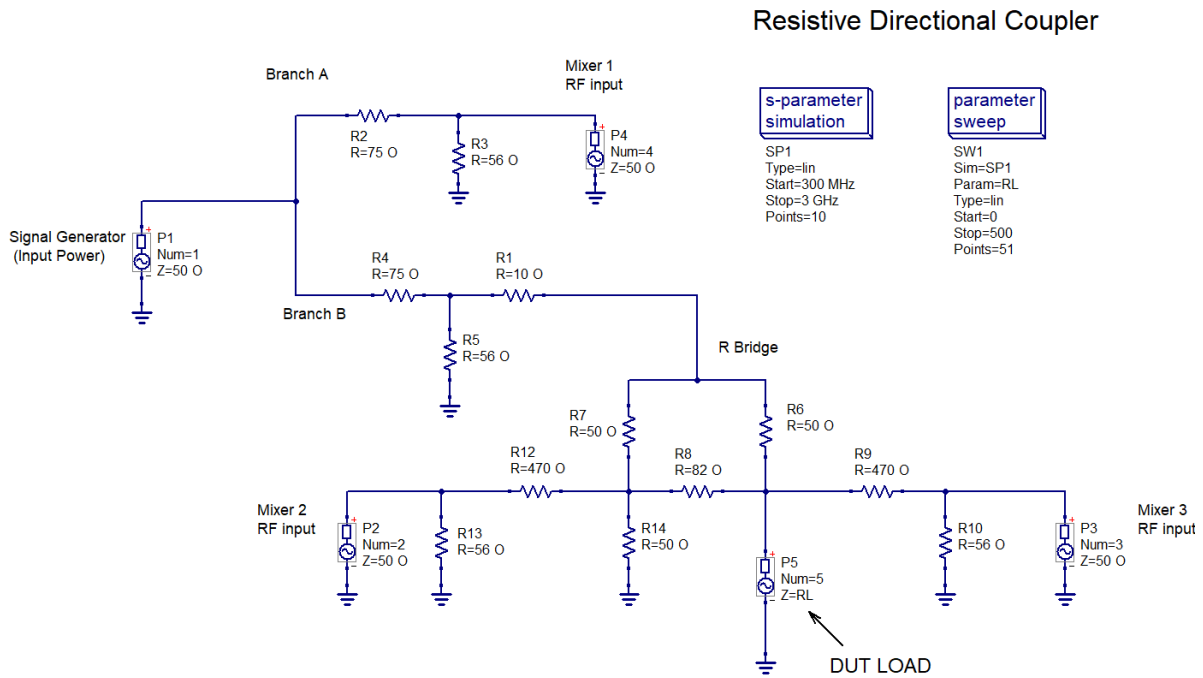


Figura 103: Arreglo propuesto en la interfaz del simulador QUCS [10].

Se realiza un barrido en la banda de frecuencias de UHF (de 300 MHz a 3 GHz) y se muestra la variación obtenida de los parámetros S con una carga  $Z_L = R_L$  que va desde cortocircuito a 500  $\Omega$ . Si se grafica la expresión 85 se observa cómo es la excursión de esta diferencia de parámetros:

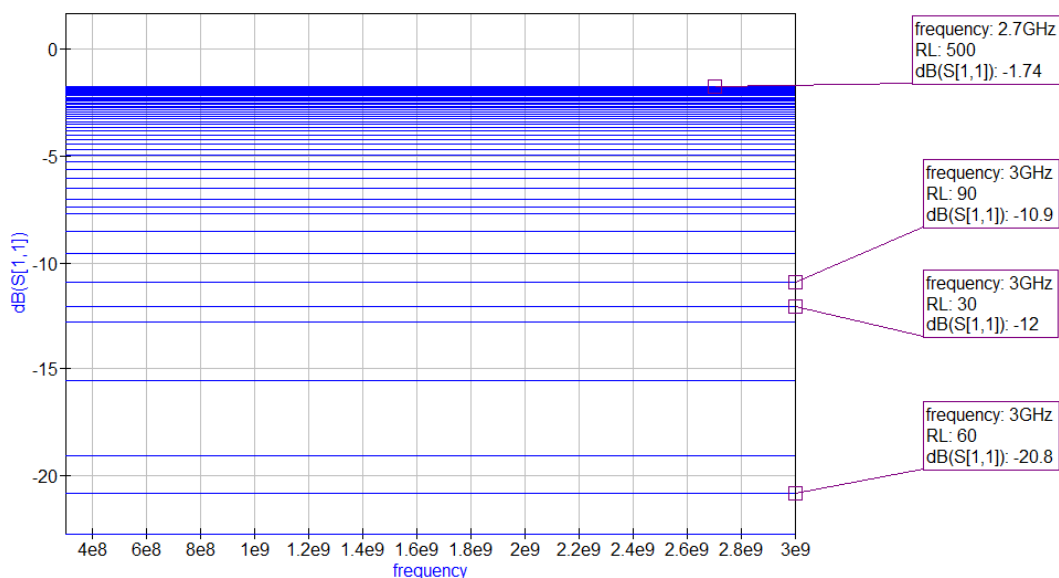


Figura 104: Coeficiente de reflexión obtenido por el acoplador.

Cuando el sistema bajo prueba presenta adaptación, la curva es nula. Cuando presenta un corto circuito, el coeficiente vale  $-1$ , y cuando presenta una carga grande, es decir, casi un circuito abierto respecto de  $Z_o$ , el coeficiente tiende a  $+1$ .

### Coeficiente de reflexión

Sea un dispositivo bajo prueba de al menos dos puertos, su coeficiente de reflexión se define como:

$$S_{11}(DUT) = \Gamma = \frac{Z_L - Z_o}{Z_L + Z_o} \tag{86}$$

Se realiza entonces una simulación con los mismos parámetros que el caso anterior: barrido de UHF y carga resistiva variable desde corto circuito a  $500 \Omega$ :

### Reflection Coeff. Measurement

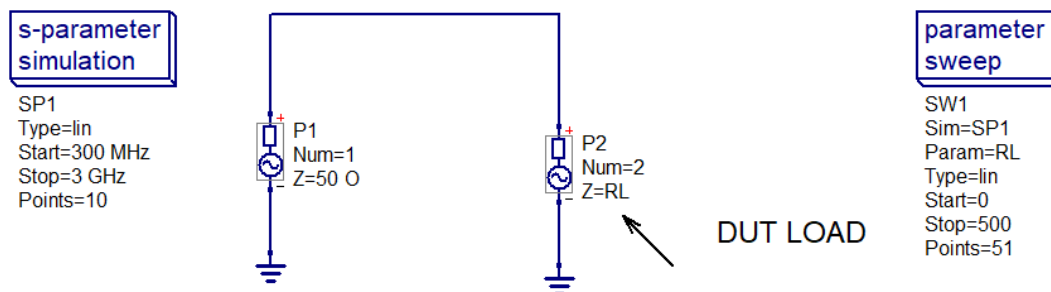


Figura 105: Simulación del coeficiente de reflexión.

Se grafica el parámetro  $S_{11}$ (dB) y se lo compara con la expresión obtenida teóricamente, ambos en función de la carga:

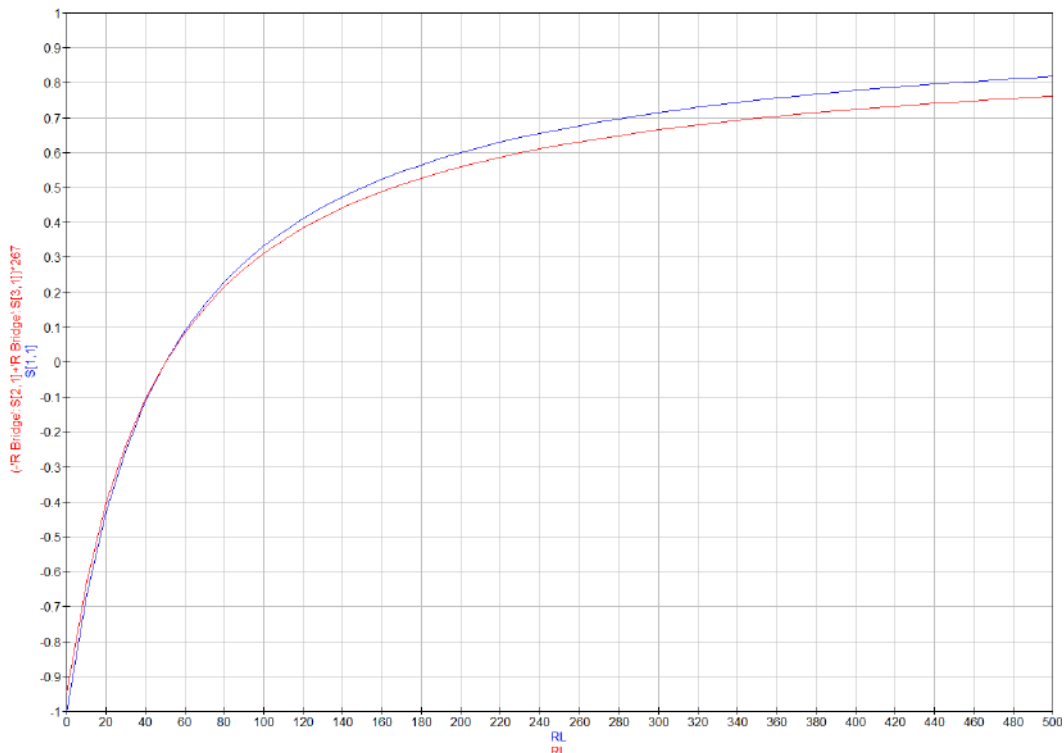


Figura 106: Gráfica  $\Gamma$  real vs  $\Gamma$  medido por el acoplador en función de la carga.

Se observa que el comportamiento entre las curvas es el mismo y parecerían cruzar por los tres puntos más importantes: corto circuito, adaptación y circuito abierto. No obstante, y por simple inspección, se ajusta el coeficiente de la expresión 85 de tal manera de que la curva que se obtiene a partir del acoplador sea esencialmente el coeficiente de reflexión verdadero. De esta manera queda finalmente:

$$\Gamma = 286 \cdot (S_{4a} - S_{4b}) \quad (87)$$

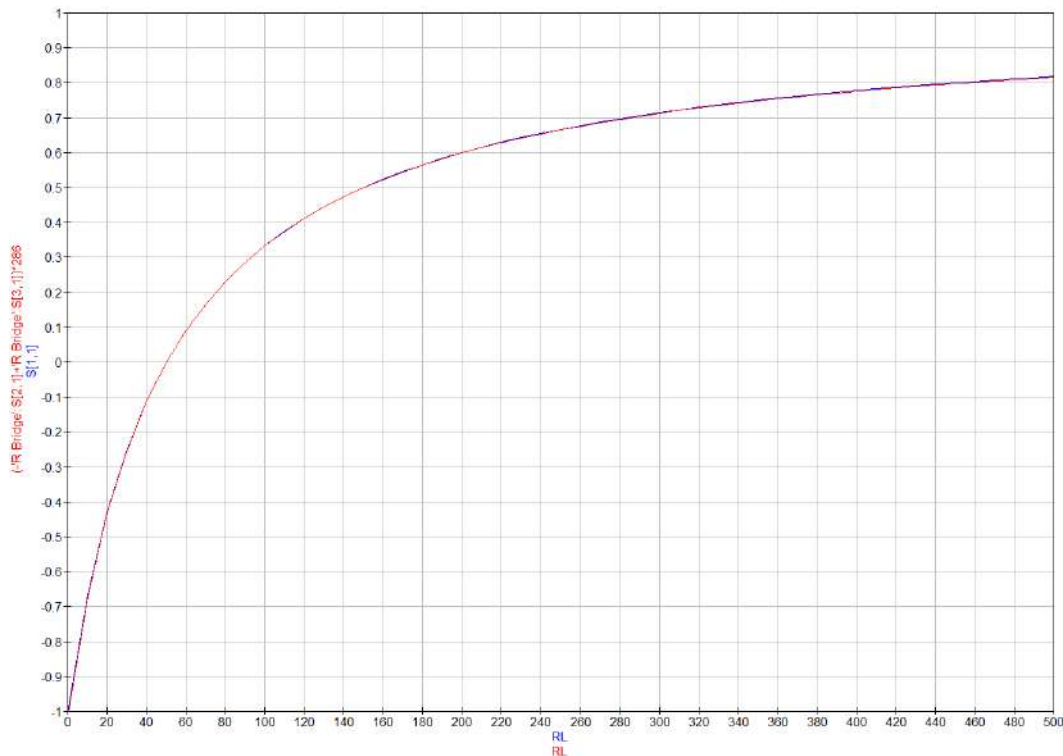


Figura 107: Gráfica  $\Gamma$  real vs  $\Gamma$  ajustado, medido por el acoplador en función de la carga.

## Resultados teóricos

En las siguientes tablas se resumen todos los parámetros del acoplador direccional propuesto y el valor en dB para los tres casos de carga: corto circuito (CC), adaptación ( $Z_o = 50 \Omega$ ) y circuito abierto (CA):

En las siguientes tablas se resumen todos los parámetros del acoplador direccional propuesto y el valor en dB para los tres casos de carga: corto circuito (CC), adaptación ( $Z_o = 50 \Omega$ ) y circuito abierto (CA):

S Parameter	CC(dB)	Zo(dB)	CA(dB)
S11	-75	-38,5	-33
S14a	-49	-44,7	-42
S14b	-inf	-44,7	-39
S13	-11,7	-11,6	11,5
S12	N/A	-19,2	-24

Tabla 7: Parámetros S1- del acoplador.



S Parameter	CC(dB)	Zo(dB)	CA(dB)
S4a1	-49	-44,7	-42
S4a4a	-51	-49,9	-49
S4a4b	-inf	-59,2	-54
S4a3	-60,7	-56,4	-54
S4a2	N/A	-33,7	-38,5

Tabla 8: Parámetros S4a- del acoplador.

S Parameter	CC(dB)	Zo(dB)	CA(dB)
S4b1	-inf	-44.7	-39.6
S4b4a	-inf	-59.2	-54
S4b4b	-68.4	-49.9	-45.3
S4b3	-inf	-56.4	-51.2
S4b2	N/A	-25.6	-30.4

Tabla 9: Parámetros S4b- del acoplador.

S Parameter	CC(dB)	Zo(dB)	CA(dB)
S31	-11,7	-11,6	-11,5
S34a	-60,7	-56,4	-54
S34b	-inf	-56,4	-51
S33	-16,4	-16,5	-16,5
S32	N/A	-30,9	-36

Tabla 10: Parámetros S3- del acoplador.

S Parameter	CC(dB)	Zo(dB)	CA(dB)
S21	N/A	-19,2	-24,1
S24a	N/A	-33,7	-39
S24b	N/A	-25,6	-30,4
S23	N/A	-30,9	-36
S22	-inf	-41,6	0

Tabla 11: Parámetros S2- del acoplador.

### 9.2.3. Hipótesis

Los resultados obtenidos indican que podría construirse un arreglo de resistores de tamaño pequeño de manera tal de poder conseguir un acoplador direccional capaz de transmitir potencia a una carga y poder extraer información de su puerto mediante la medición de su coeficiente de reflexión.

No obstante, hay que tener en cuenta que al tratarse de un análisis en donde las componentes son puramente resistivas, los resultados son únicamente de magnitud y, por lo tanto, constantes a lo largo de la banda de frecuencias de UHF. Además de esto, se espera que la longitud de todo el arreglo sea mucho menor respecto a la menor longitud de onda (100 mm) para despreciar pérdidas debidas a, por ejemplo, el efecto skin o pérdidas parásitas debido a las líneas de transmisión que se utilizarían para cada conexión de nodo o rama, en un circuito impreso con placas dieléctricas y laminados de cobre.

#### 9.2.4. Implementación en VNA

##### Diseño

Recordando que el estudio del acoplador resistivo direccional surge a partir de la necesidad de desarrollar un VNA de gran ancho de banda y de menor tamaño, se presenta la etapa del acoplador direccional resistivo, acompañado de los conectores SMA correspondientes a la entrada y salida de la carga. Justo debajo del arreglo resistivo se encuentra el banco de Mixers junto a una etapa de pre-filtrado hacia la etapa de frecuencia intermedia que está en la parte superior derecha de la placa.

Prestándole atención a la sección del arreglo resistivo, vemos que es una versión acomodada del puente del prototipo 1, ordenado los componentes de tal manera de lograr la mejor adaptación a la próxima etapa de Mixers. Además, está diseñada teniendo en cuenta algunas de las mejores prácticas en términos de diseño de RF y compatibilidad electromagnética: Líneas de transmisión de  $50\ \Omega$  coplanares con blindaje, caminos de corrientes de retorno adecuados y planos de masa interconectados mediante varias vías.

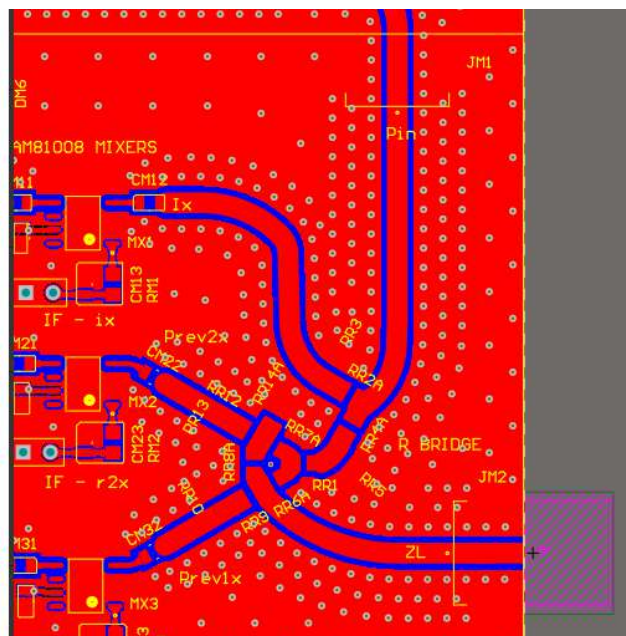


Figura 108: Layout del acoplador direccional resistivo en VNA.

El puente mide aproximadamente  $13\text{mm} \times 13\text{mm}$ . Las líneas de transmisión hacia los Mixers 2 y 3 miden  $6,2\text{m}$ . La línea hacia la carga mide  $22,4\text{mm}$ . La línea de la potencia de entrada mide  $31\text{mm}$  y la línea hacia el Mixer de referencia mide  $20\text{mm}$ . En todos los casos se tienen en cuenta a las curvaturas de la línea.



Figura 109: Vista tridimensional del acoplador direccional resistivo en VNA.

### 9.2.5. Simulaciones con método de elementos finitos

Para corroborar el funcionamiento del acoplador se realizan dos simulaciones: una en QUCS [10], en el cual se modelan las líneas de transmisión y otra en el simulador CST Studio [9], de mayor complejidad, en el cual se simula la sección directamente del diseño de PCB de manera tridimensional bajo métodos finitos y resolviendo las ecuaciones de Maxwell [4]. Esta última nos brindaría resultados más cercanos a la realidad ya que tiene en cuenta las vías cercanas, las líneas de transmisión con sus longitudes y desviaciones verdaderas, los *pads* de los resistores, la capa dieléctrica, entre otros.

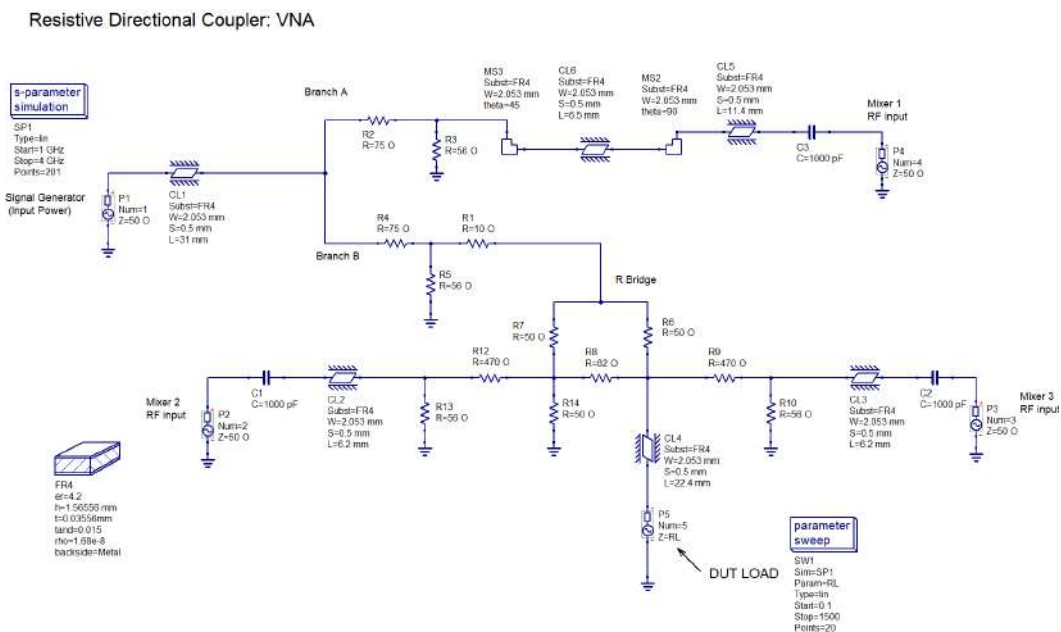


Figura 110: Simulación en QUCS [10].

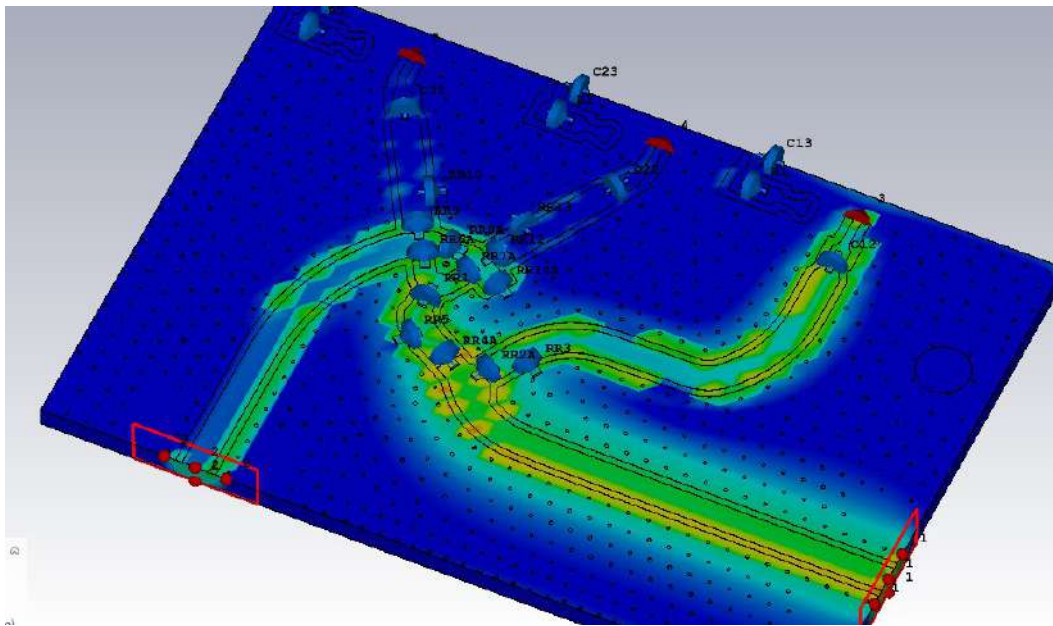


Figura 111: Simulación en CST Studio [9].

En la figura siguiente se observa la componente del campo eléctrico y el efecto de blindaje de las vías para minimizar su desparramo en el resto de la placa.

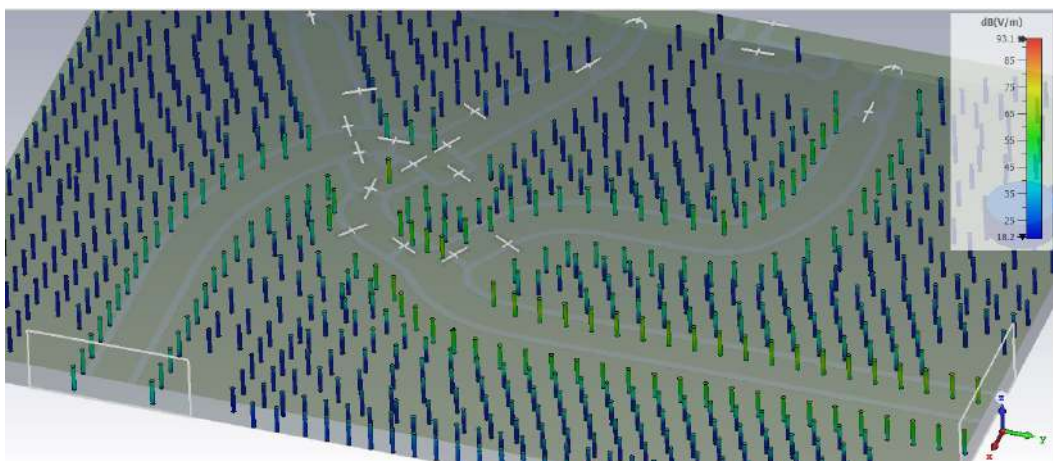


Figura 112: Simulación en CST Studio [9].

En ambas simulaciones se realiza un barrido en frecuencia desde 1 GHz a 4 GHz ya que nos interesa más el comportamiento de la placa en alta frecuencia para una variación de la carga de  $0,1 \Omega$  a  $1500 \Omega$ :

### 9.2.6. Parámetros S

Las siguientes gráficas muestran cada uno de los parámetros S del acoplador, tanto de la simulación del QUCS como de la simulación del CST:

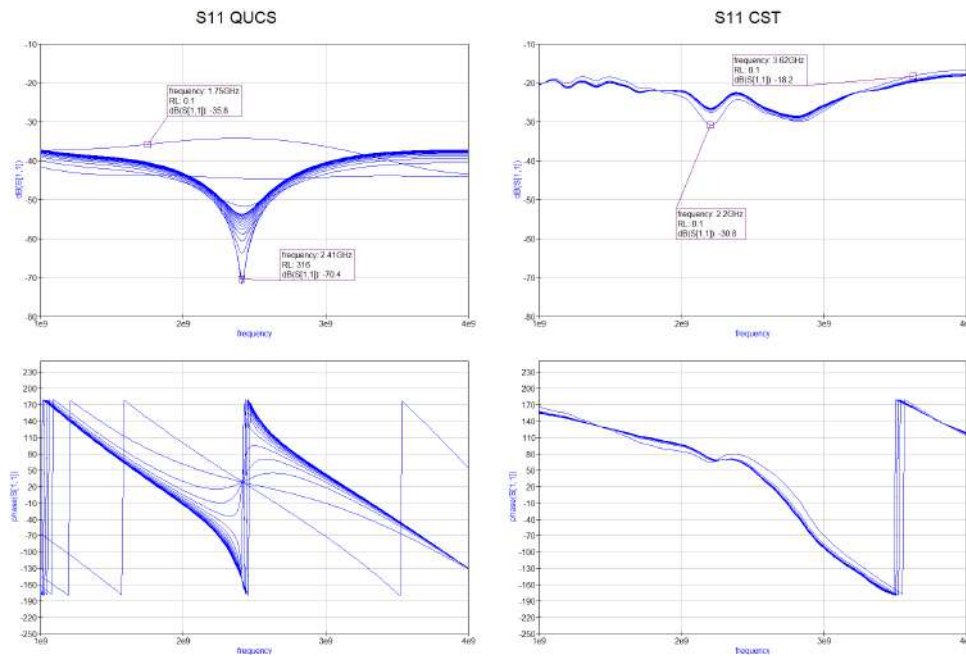


Figura 113: Comparación de S11.

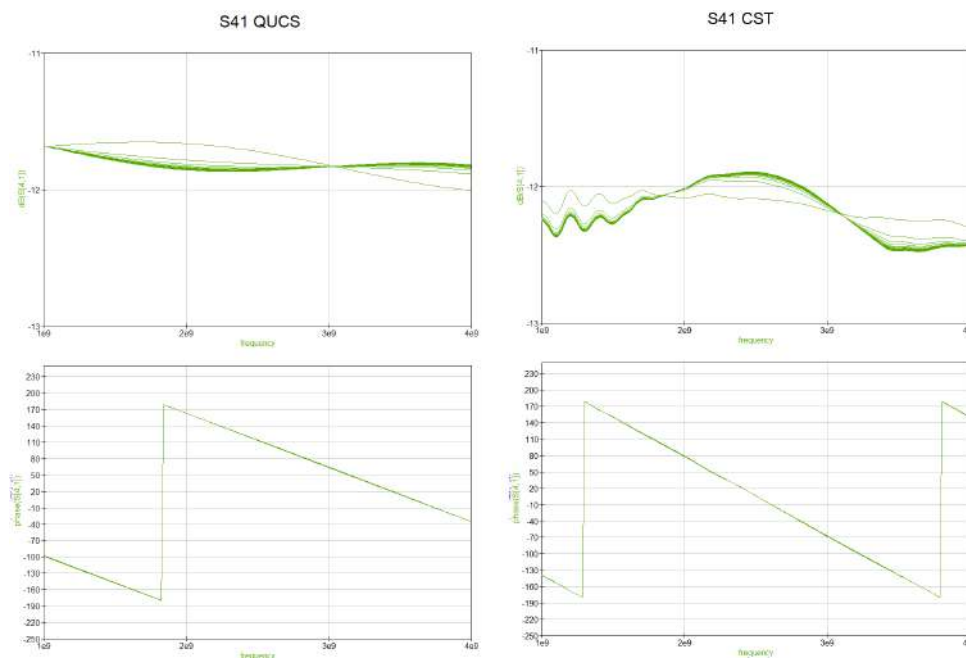


Figura 114: Comparación de S41.

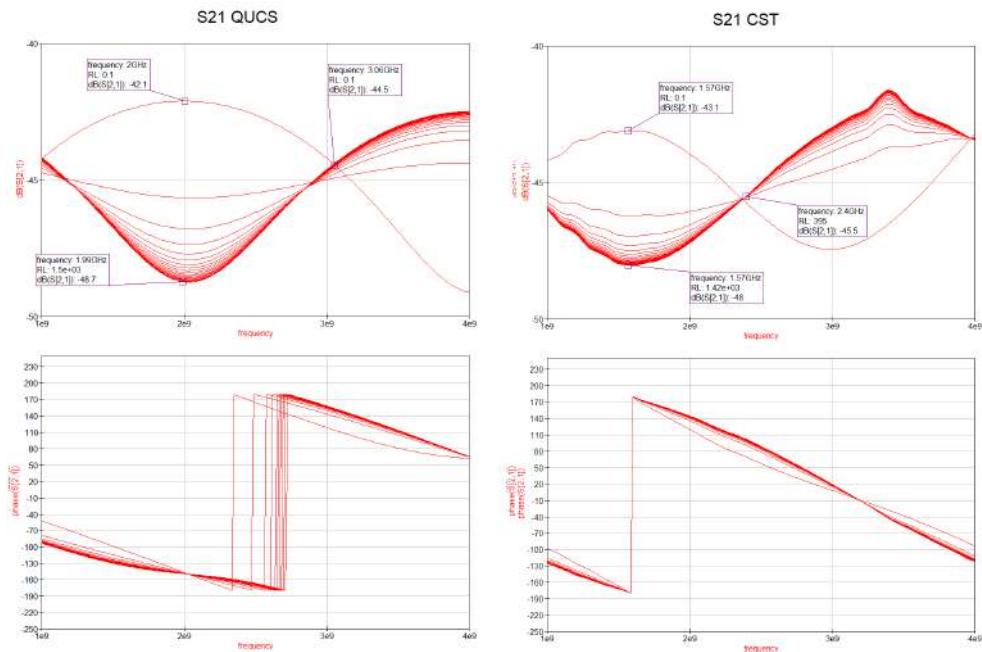


Figura 115: Comparación de S21.

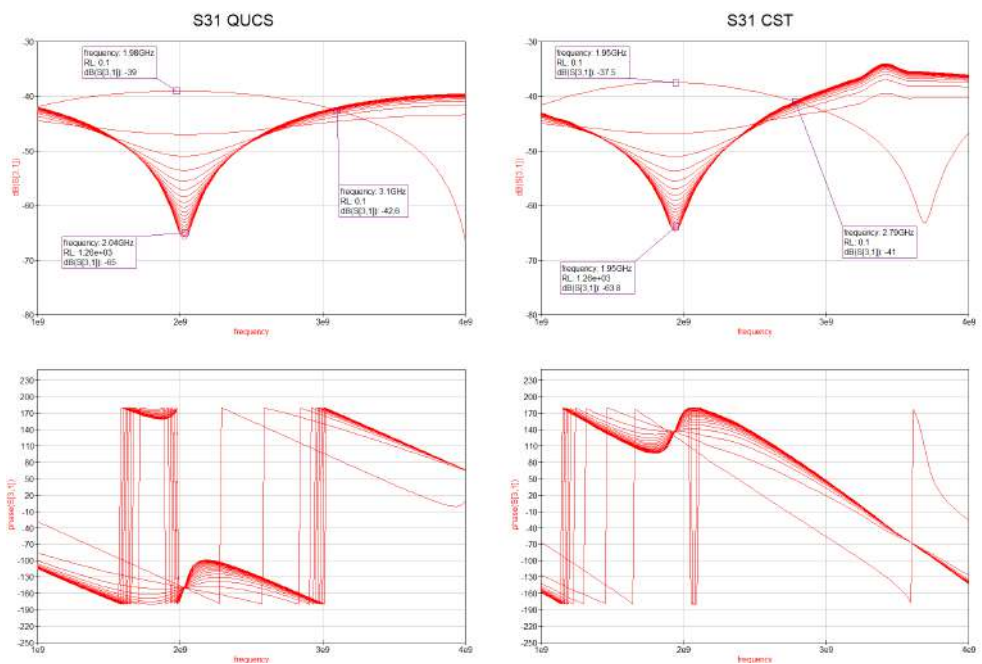


Figura 116: Comparación de S31.

### Coeficiente de reflexión

Al realizar la diferencia entre los parámetros S31 y S21, hemos de esperar un término proporcional al coeficiente de reflexión. En la figura siguiente se comparan directamente los resultados obtenidos tanto en QUCS como en CST de este parámetro.

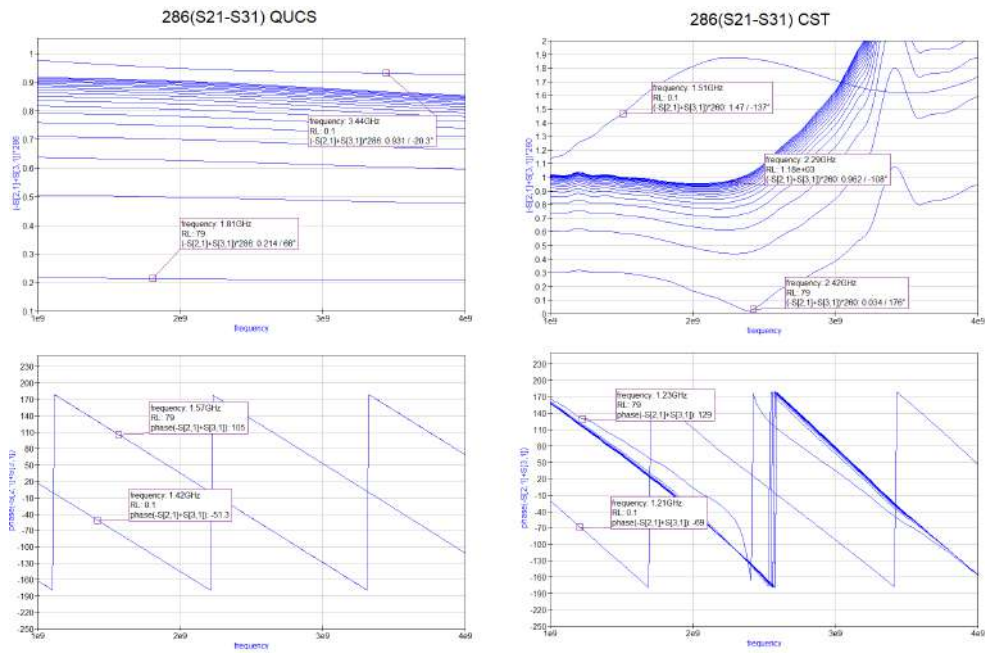


Figura 117: Comparación de coeficientes de reflexión.

Mientras que el comportamiento en QUCS es muy similar al prototipo presentado en la sección anterior, CST nos brinda información un tanto más compleja pero cuyos resultados parecen seguir la misma lógica proporcional del coeficiente de reflexión.

Es más evidente si se grafica para un conjunto de frecuencias en función de la carga tales parámetros y comparándolos con el coeficiente ideal:

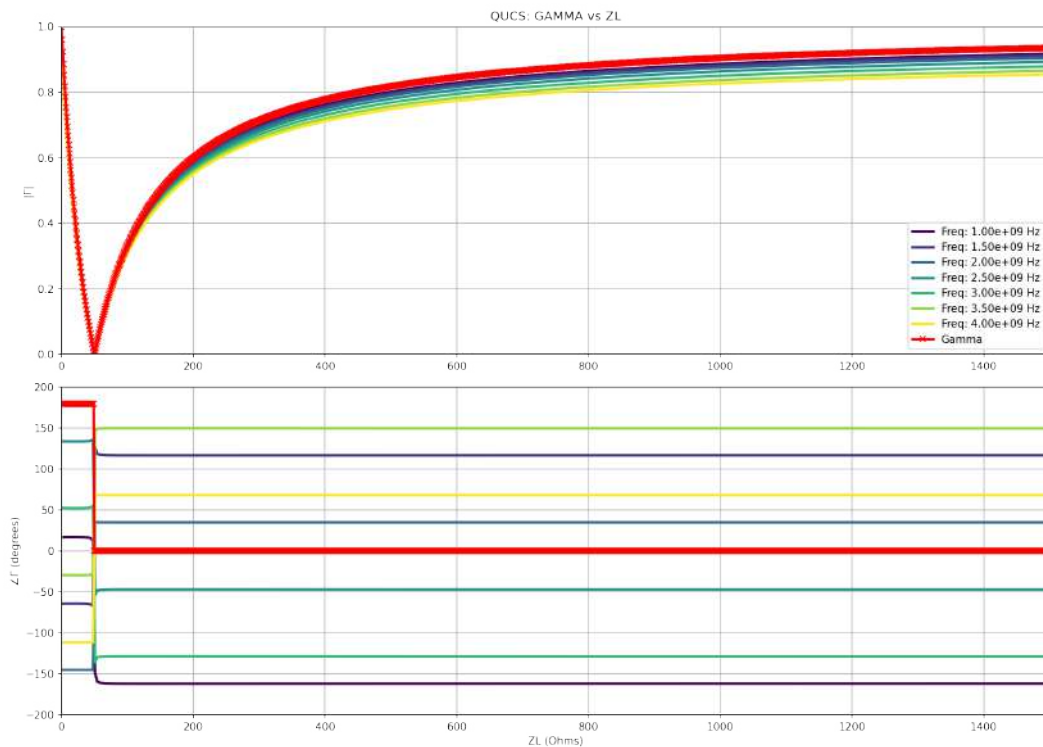


Figura 118: Coeficientes de reflexión: simulación en QUCS.

El resultado anterior muestra cómo a mayor frecuencia la atenuación empieza a predominar. Por

otro lado, y debido a las líneas de transmisión, hay desfases diferentes para cada frecuencia. Aún así, se mantiene la diferencia de  $180^\circ$  en relación con la carga, es decir, el signo del coeficiente de reflexión.

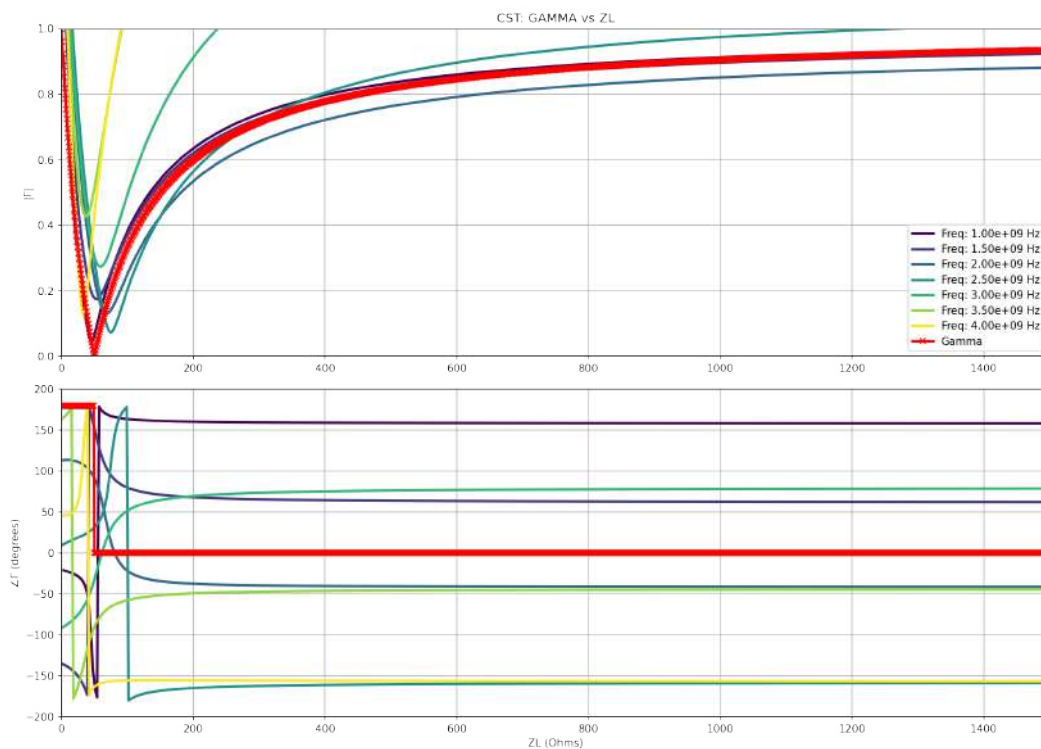


Figura 119: Coeficientes de reflexión: simulación en CST.

Los resultados de CST predicen el comportamiento típico y de gran atenuación que hay en un circuito de micro ondas realizable. Se puede observar como a diferentes frecuencias los desfases y el error en la magnitud son diferentes pero siguen la misma característica que la curva ideal. No obstante, esto debería mitigarse empleando una buena técnica de calibración basada en el conocimiento de los tres puntos principales de medición: Circuito Abierto, Corto Circuito e Impedancia de  $50 \Omega$ , armando una matriz de error para cada una de las frecuencias, corrigiendo así el error particular tanto en fase como en amplitud.





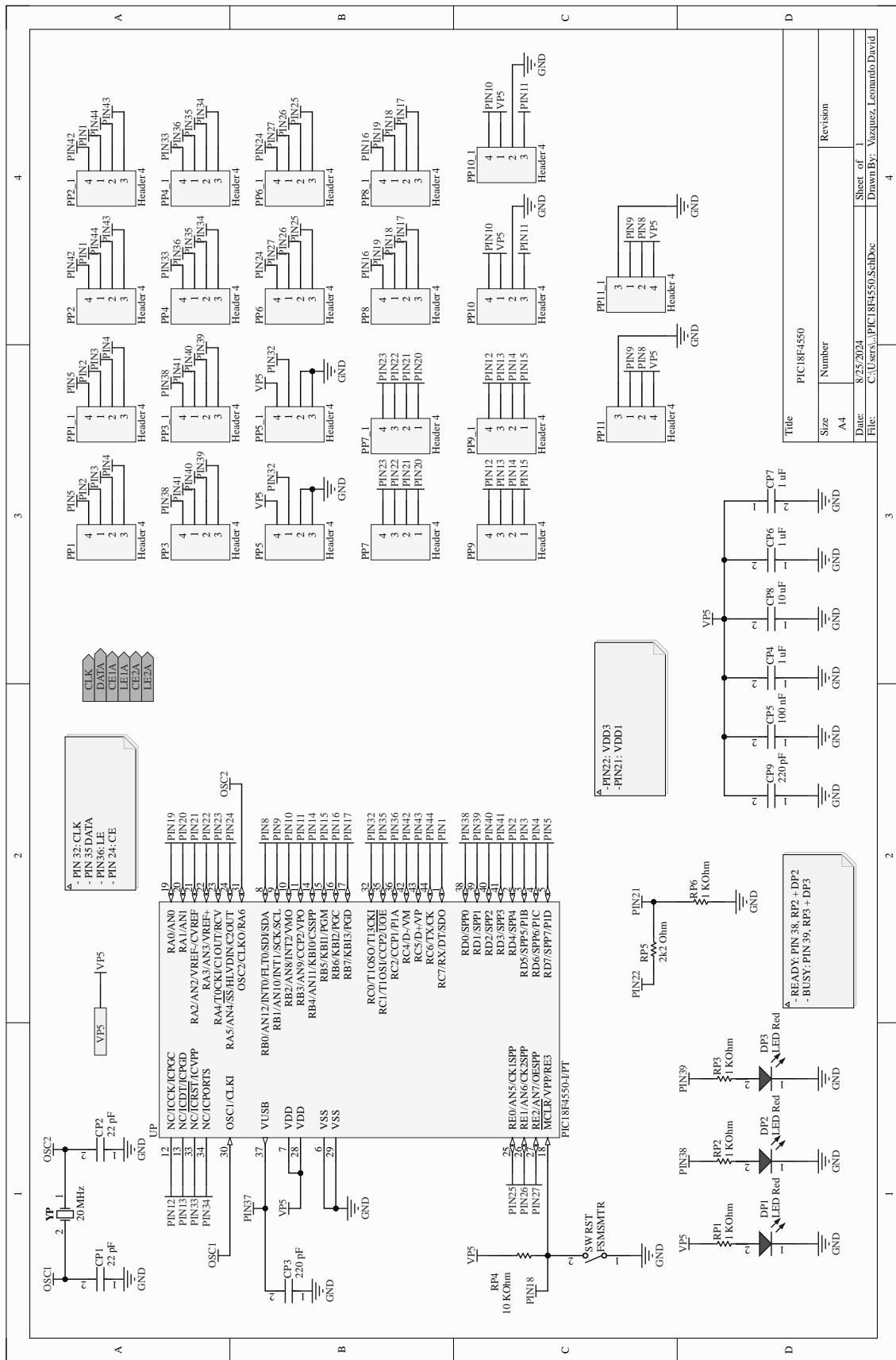


Figura 121: Microcontrolador PIC18F4550 I/PT.

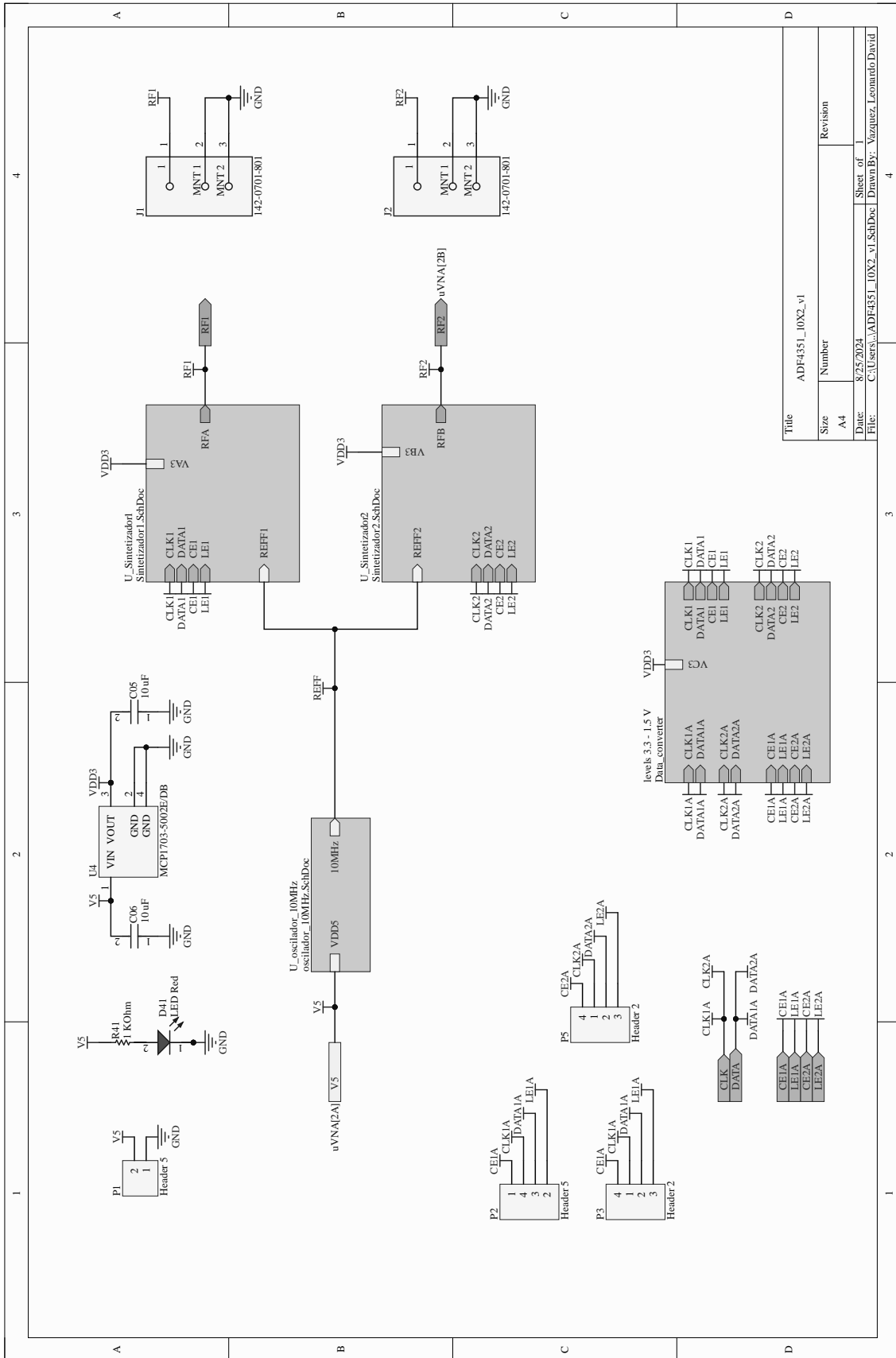


Figura 122: Diagrama de Generadores de RF.

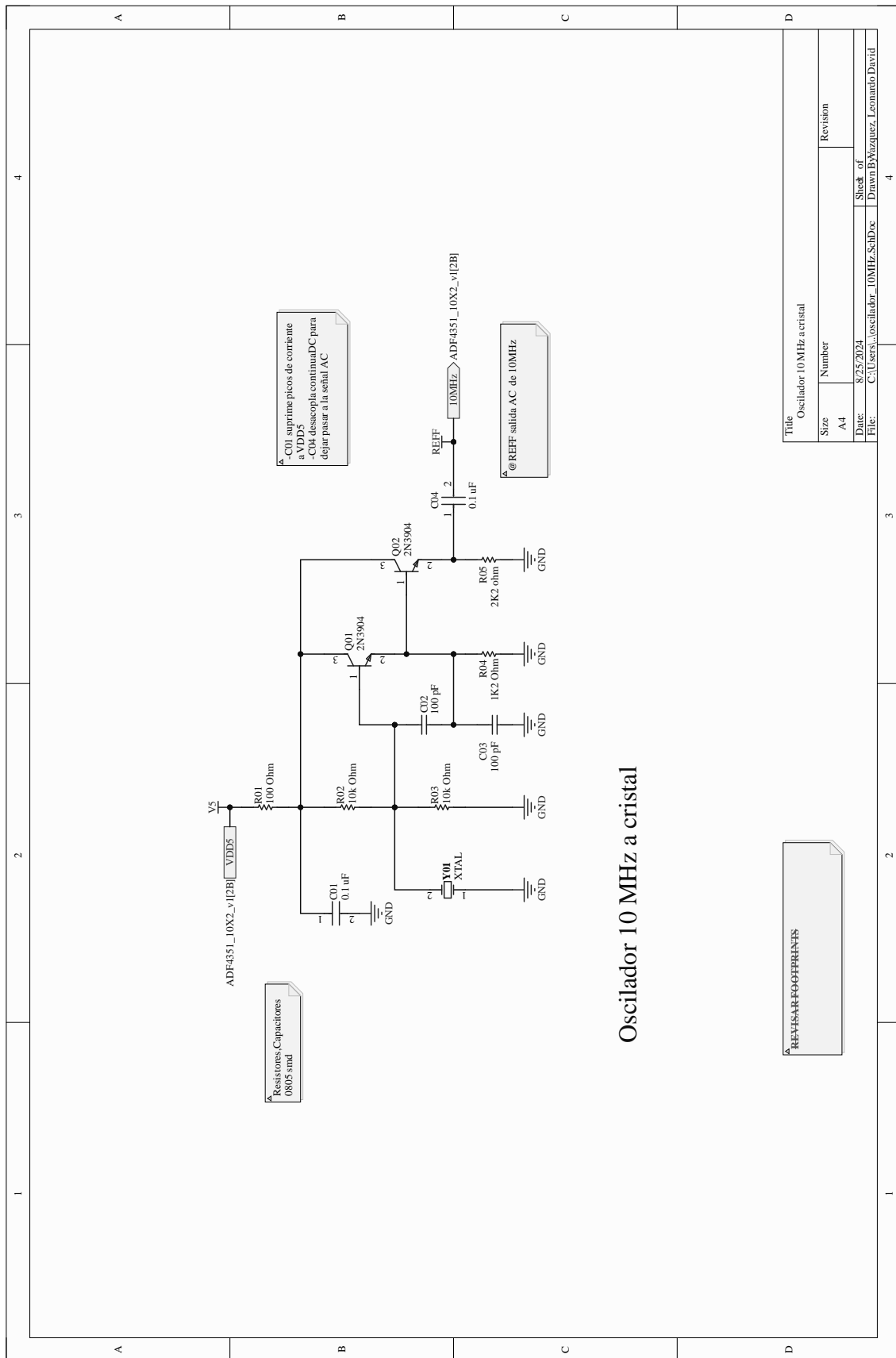


Figura 123: Circuito Oscilador de Referencia.

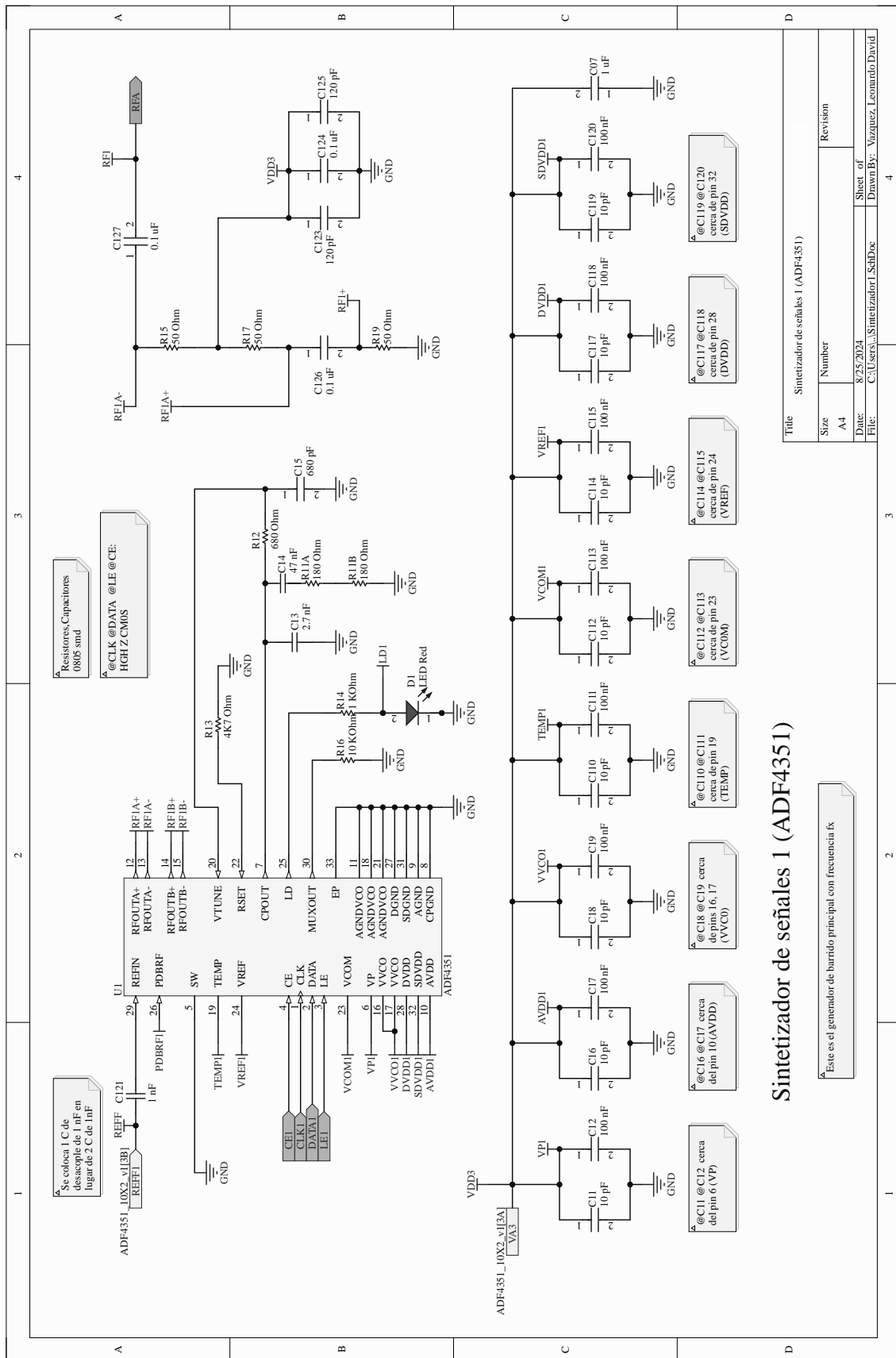


Figura 124: Sintetizador de señales de RF GEN-1.

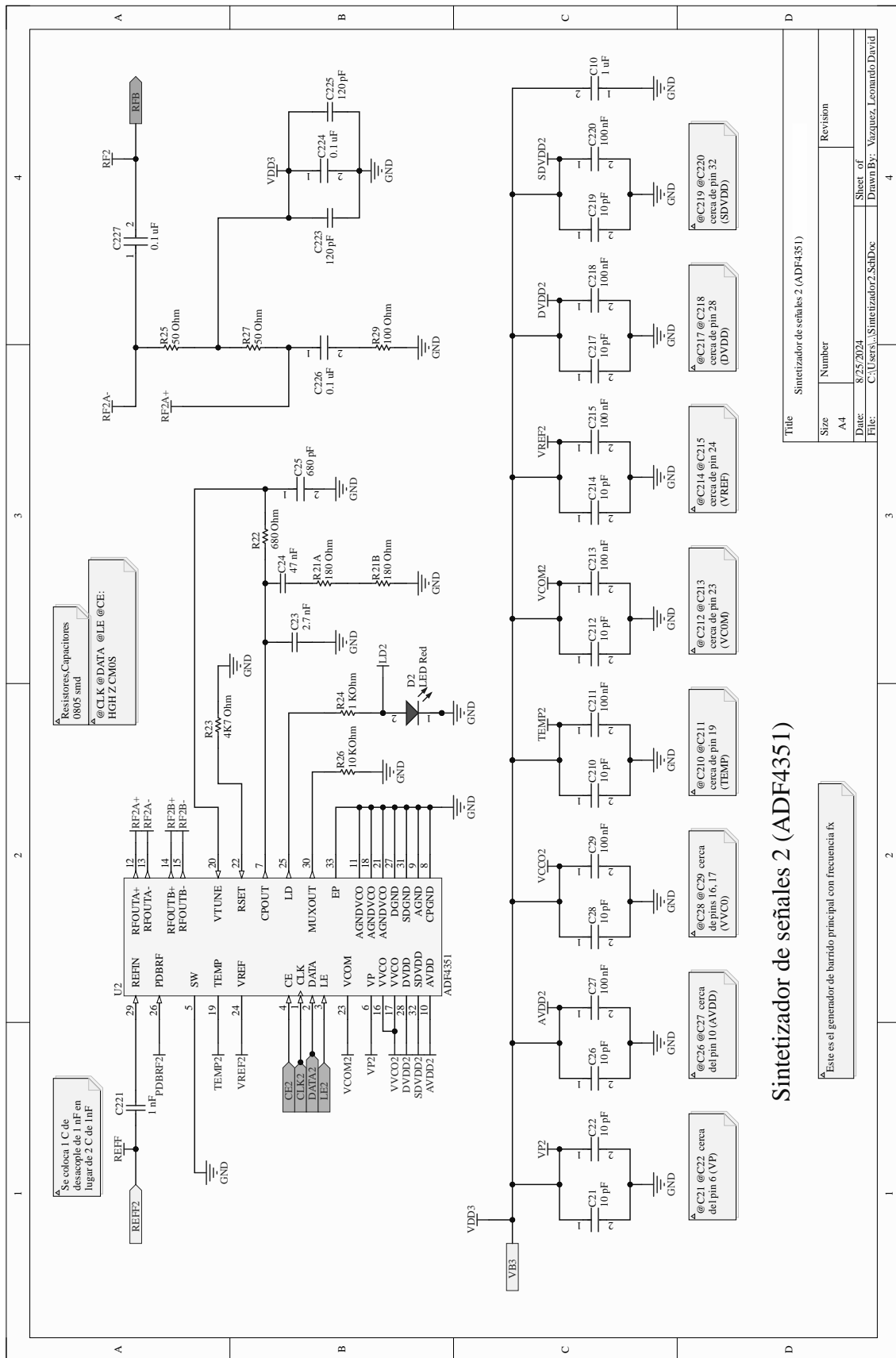


Figura 125: Sintetizador de señales de RF GEN-2.

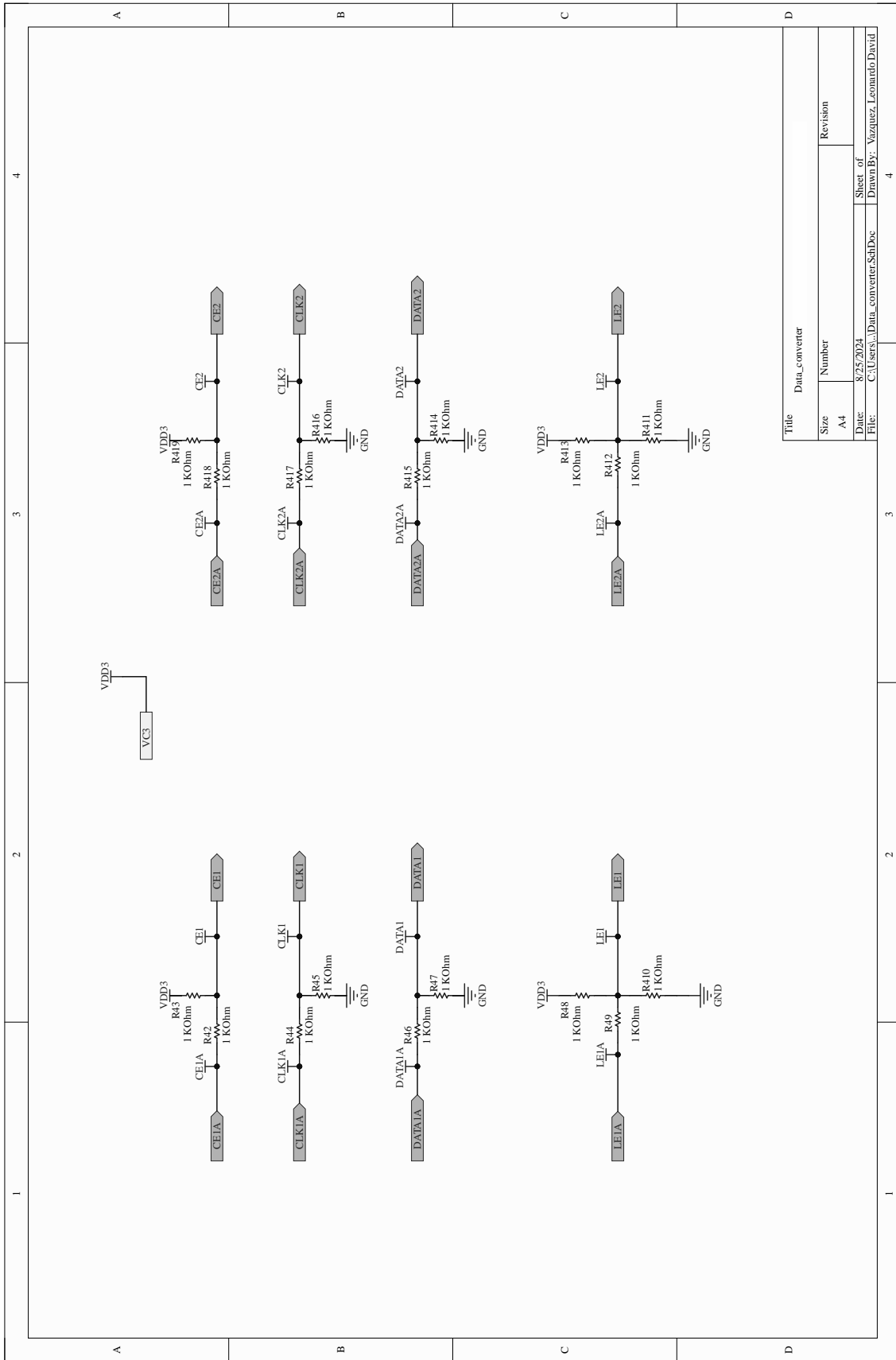


Figura 126: Conversores de niveles.

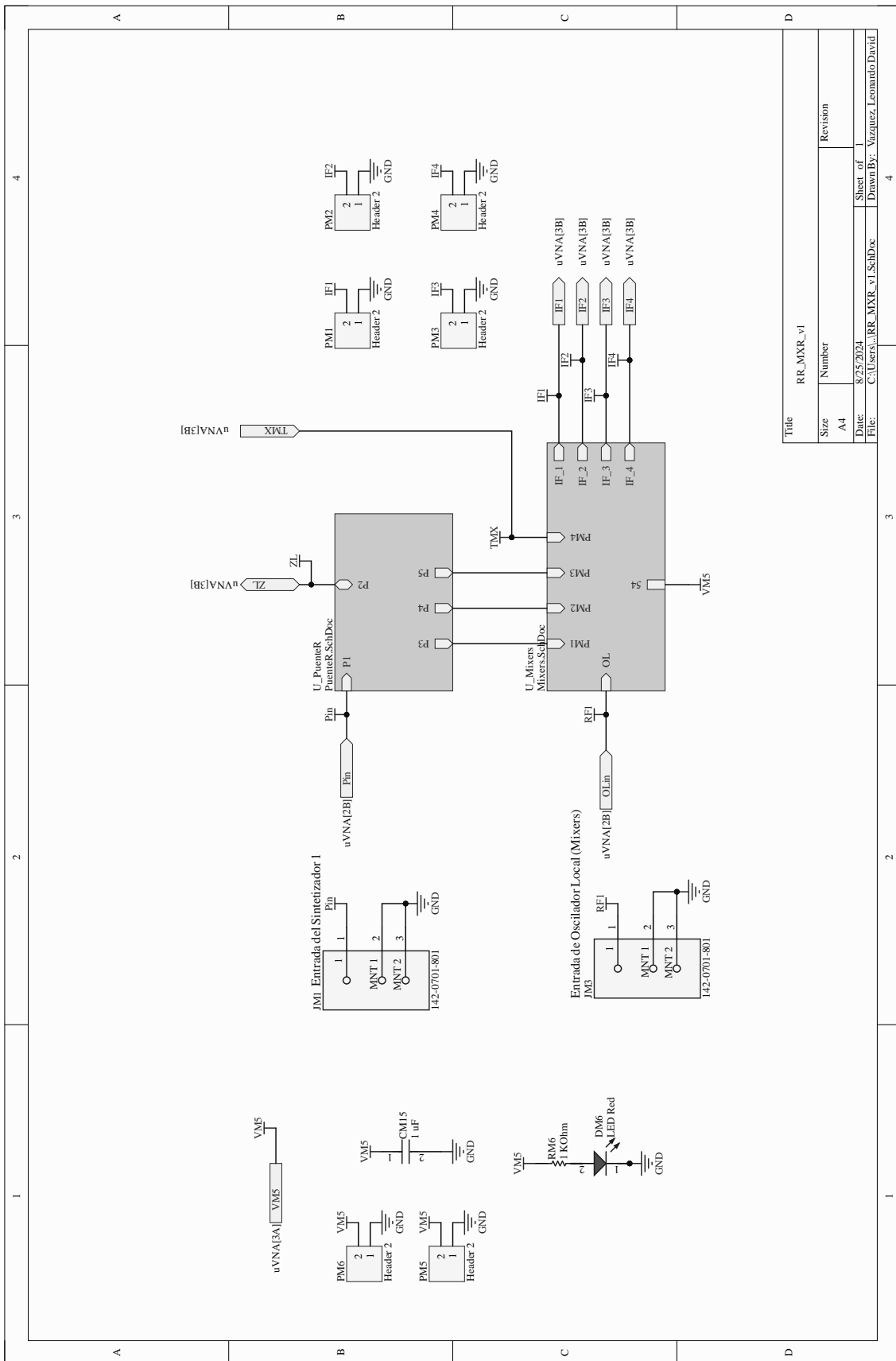


Figura 127: Diagrama de Receptor Heterodino.



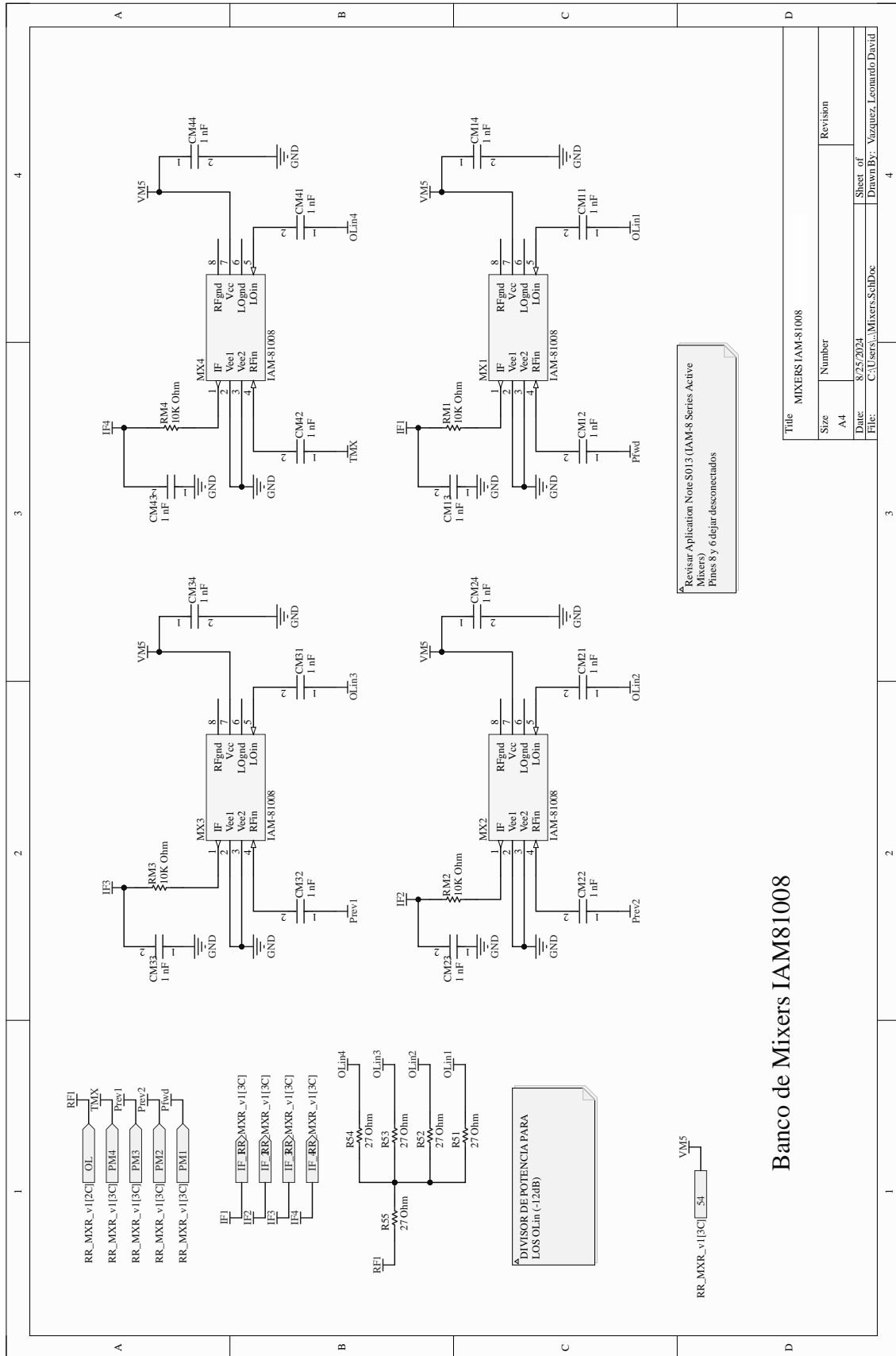
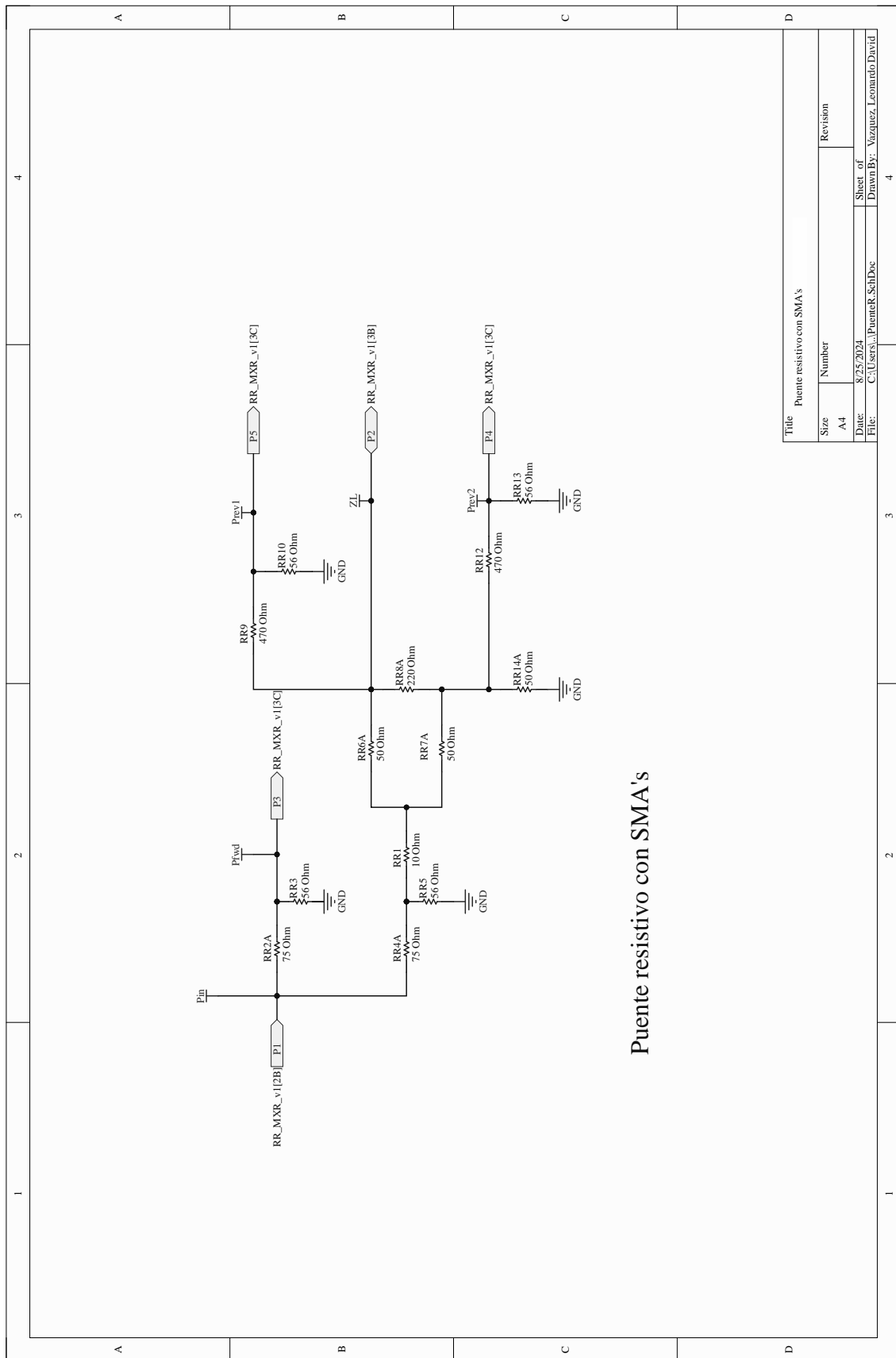
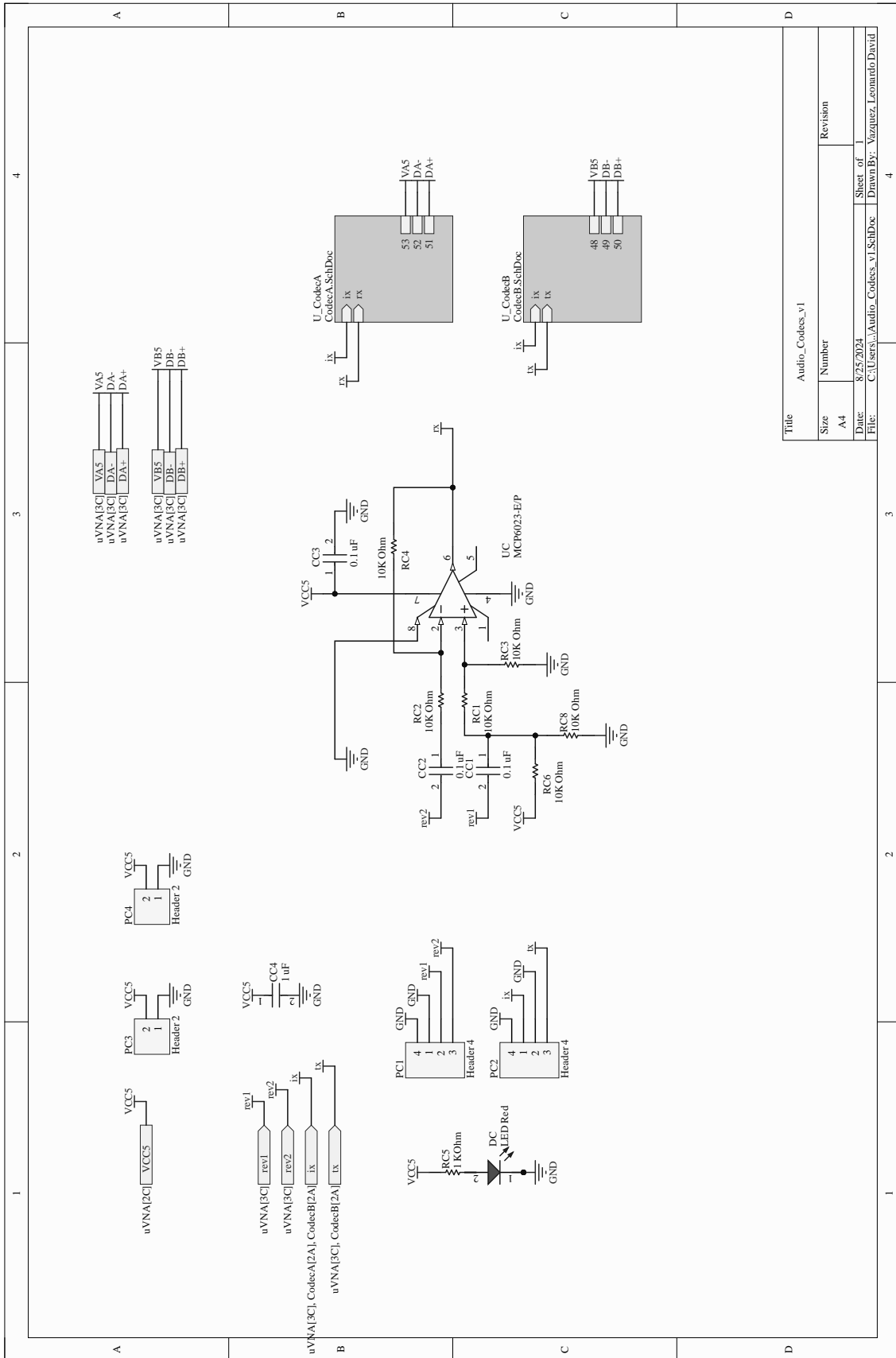


Figura 128: Banco de Mixers.



Title	
Puente resistivo con SMA's	
Size	Number
A4	
Revision	
Date:	8/25/2024
File:	C:\Users\...PuenteR_Sch1Doc
Sheet of	4
Drawn By:	Vizquez, Leonardo David

Figura 129: Acoplador Direccional Resistivo.



Title		Revision	
Size	A4	Number	
Date:	8/25/2024	Sheet of	4
File:	C:\Users\...Audio_Codexes.v1.SchDoc	Drawn By:	Vizquez, Leonardo David

Figura 130: Amplificador restador.

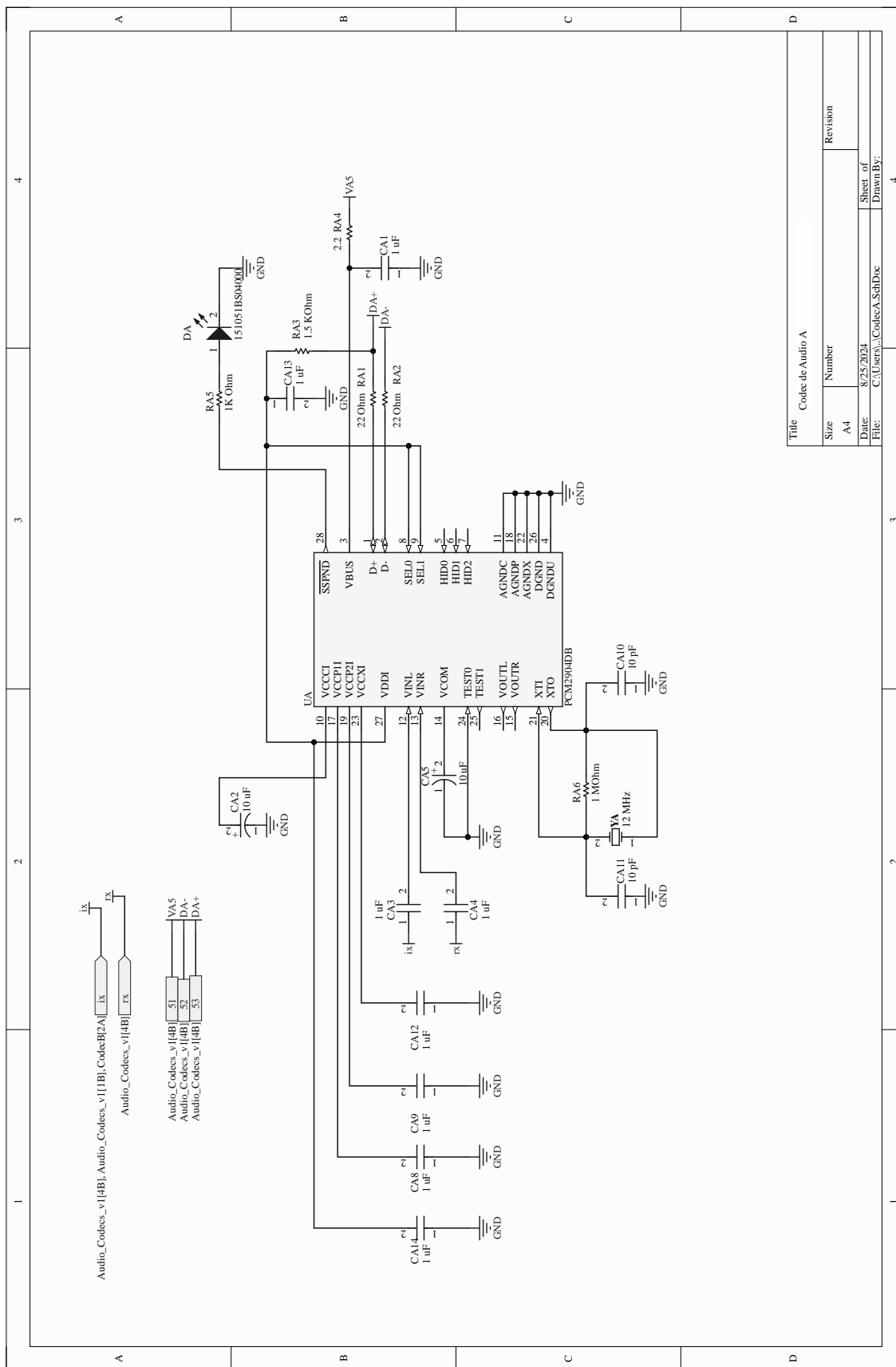


Figura 131: Codec de Audio A: S11/S22.

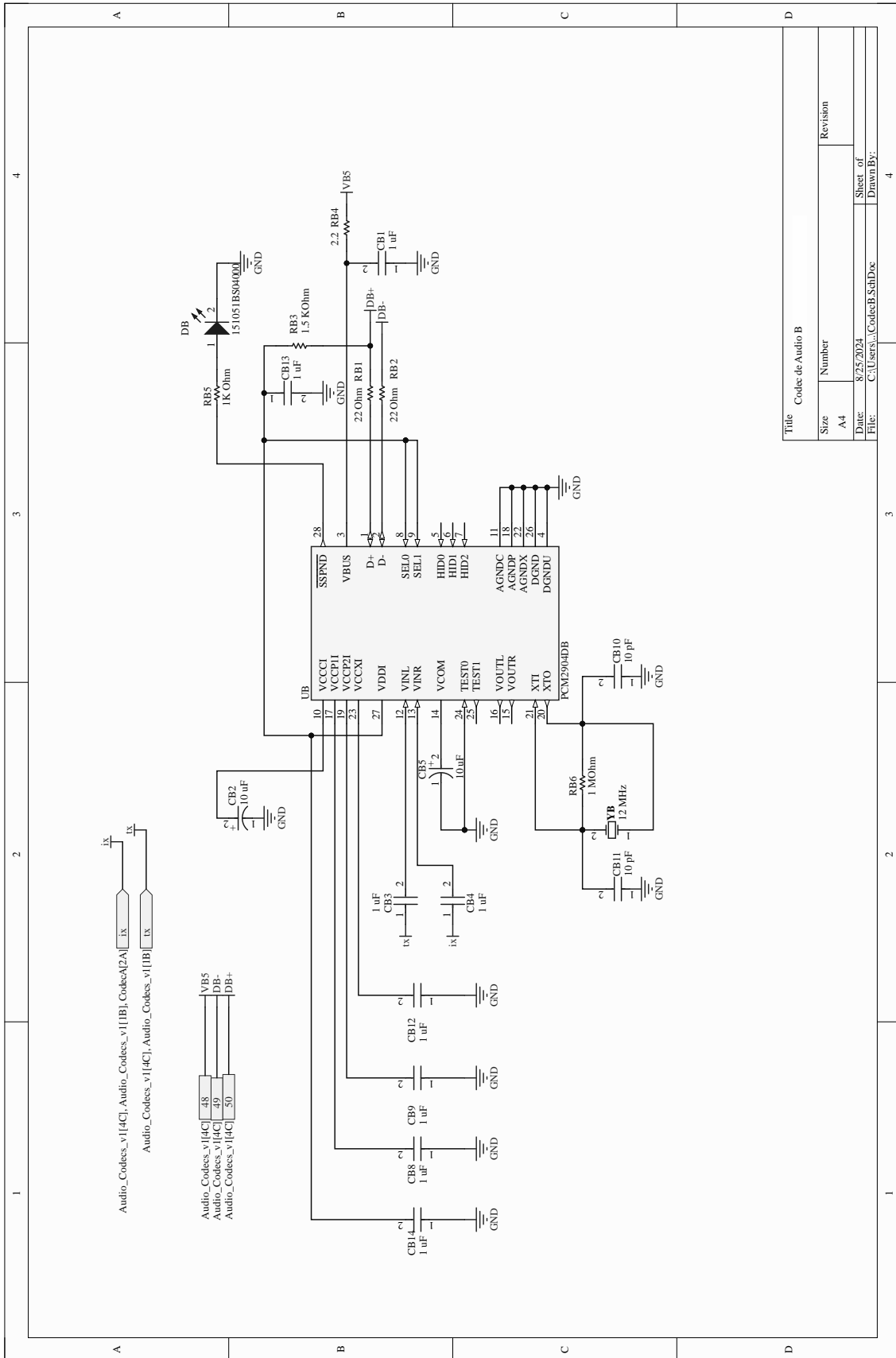


Figura 132: Codec de Audio B: S21/S12.

## 9.4. Lista de componentes

PIC18F4550			
Designator	Name	Value	Quantity
UP	PIC18F4550 -I/PT	44 PINS	1
RP1, RP2, RP3, RP6	RES SMD	1 K OHM	4
RP4	RES SMD	10 K OHM	1
RP5	RES SMD	2 K 2 OHM	1
DP1, DP2, DP3	LED SMD	-	3
YP	XTAL	20 MHZ	1
CP1, CP2, CP3, CP9	CAP SMD	220 pF	4
CP4	CAP SMD	-	-
CP5	CAP SMD	100 nF	1
CP6, CP7	CAP SMD	1 uF	2
CP8	CAP SMD	10 uF	1
SW RST	SWITCH THR HOLE	-	1
PP (1-11)	HEADER X4	4 PINS	22

Tabla 12: Componentes: Microcontrolador.

R BRIDGE + MIXERS			
Designator	Name	Value	Quantity
MX1, MX2, MX3, MX4	IAM-81008	8 PINS	4
RM1, RM2, RM3, RM4	RES SMD	10 K OHM	4
RM6	RES SMD	1 K OHM	1
CM15	CAP SMD	1 uF	1
CM11, CM12, CM13, CM14	CAP SMD	1 nF	16
CM21, CM22, CM23, CM24			
CM31, CM32, CM33, CM34			
CM41, CM42, CM43, CM44			
R51, R52, R53, R54, R55	RES SMD	27 OHM	5
RR1	RES SMD	10 OHM	1
RR2A, RR4A	RES SMD	75 OHM	2
RR3, RR5, RR10, RR13	RES SMD	56 OHM	4
RR6A, RR7A, RR14A	RES SMD	50 OHM	3
RR8A	RES SMD	220 OHM	1
RR9, RR12	RES SMD	470 OHM	2
DM6	LED SMD	-	1
JM2, JM4	SMA	-	2
PM5, PM6	HEADER X2	2 PINS	2

Tabla 13: Componentes: Receptor Heterodino.

ADF4351 X2			
Designator	Name	Value	Quantity
U1, U2	ADF4351	32 PINS	2
R01	RES SMD	100 OHM	1
R01, R03, R26, R16	RES SMD	10 K OHM	2
R04	RES SMD	1 K 2 OHM	1
R05	RES SMD	2 K 2 OHM	1
R42, R32, R44, R45	RES SMD	1 K OHM	23
R46, R47, R48, R49			
R410, R411, R412, R413, R414, R415, R416, R417			
R418, R419			
R14, R24, R41			
R13, R23			
R11A, R11B, R21A, R21B	RES SMD	360 OHM	4
R12,R22	RES SMD	680 OHM	2
R15, R17, R19, R25, R27, R29	RES SMD	50 OHM	6
C11, C16, C18, C110	CAP SMD	10 pF	16
C112, C114, C117, C119			
C21, C26, C28, C210			
C212, C214, C217, C219			
C13, C23	CAP SMD	2,7 nF	2
C14, C24	CAP SMD	47 nF	2
C15, C25	CAP SMD	680 pF	2
C12, C17, C19, C111, C113, C115, C118, C120	CAP SMD	100 nF	16
C22, C27, C29, C211, C213, C215, C218, C220			
C123, C125, C223, C225	CAP SMD	120 pF	4
C02, C03	CAP SMD	100 pF	2
C04, C127, C126, C124 C227, C226, C224	CAP SMD	0,1 uF	8
C05, C07, C010	CAP SMD	1 uF	6
C121, C221	CAP SMD	1 nF	2
C06, C01, C08, C09	CAP SMD	10 uF	4
D1, D2, D41	LED SMD	-	3
Q01, Q02	2N3904	-	2
U4	MCP1703 REG	3,3 V - 500 mA	1
Y01	XTAL	10 MHZ	1
P1	HEADER X2	2 PINS	2
P2, P3, P5	HEADER X4	4 PINS	3

Tabla 14: Componentes: Generadores de RF.

AUDIO CODECS			
Designator	Name	Value	Quantity
UA, UB	PCM2904	28 PINS	2
YA, YB	XTAL	12 MHz	2
RA1, RA2, RB1, RB2	RES SMD	22 OHM	4
RA3, RB3	RES SMD	1 K 5 OHM	2
RA4, RB4	RES SMD	2,2 OHM	2
RA5, RB5 , RC5	RES SMD	1 K OHM	3
RA6, RB6	RES SMD	1 M OHM	2
RC1, RC2 RC4, RC6, RC8, RC3	RES SMD	10 K OHM	6
CA10, CA11, CB10, CB11	CAP SMD	10 pF	4
CA1, CB1, CA13, CB13, CA14, CB14, CA3, CA4, CB3, CB4, CA8, CA9, CA12, CB8, CB9, CB12	CAP SMD	1 uF	16
CC1, CC2, CC3, CC4	CAP SMD	0,1 uF	4
CA2, CA5, CB2, CB5	CAP ELECTR	10 uF / 100 V	4
DA, DB	LED THR HOLE	-	2
DC	LED SMD	-	1
PC3, PC4	HEADER X2	2 PINS	2
PC1, PC2	HEADER X4	4 PINS	2
UC	MCP6023	8 PINS	1
JA, JB	USB TYPE B SMD	4 PINS	2

Tabla 15: Componentes: Adquisición de audio.



## 9.5. Firmware en Embedded C

### 9.5.1. Librerías

```
#include <xc.h>
#include <stdlib.h>
#include <p18f4550.h>
#include <math.h>
```

### 9.5.2. Configuración

```
#pragma config PLLDIV = 5, CPUDIV = OSC1_PLL2, USBDIV = 2
#pragma config FOSC = HSPLL_HS, FCMEN = OFF, IESO = OFF
#pragma config PWRT = OFF, BOR = OFF, VREGEN = OFF
#pragma config WDT = OFF, WDTPS = 32768
#pragma config MCLRE = ON, LPT1OSC = OFF, PBAEN = OFF
#pragma config STVREN = ON, LVP = OFF, ICPRT = OFF, XINST = OFF
```

### 9.5.3. Definiciones

```
#define _XTAL_FREQ 4800000
#define LED1 LATDbits.LATD0 //PIN_D0 (pin38)
#define LED2 LATDbits.LATD1 //PIN_D1 (pin39)
#define CLK1 LATDbits.LATD7 //PIN_D7 (pin5)
#define DATA_S1 LATDbits.LATD6 //PIN_D6 (pin4)
#define LE1 LATDbits.LATD5 //PIN_D5 (pin3)
#define CE1 LATDbits.LATD4 //PIN_D4 (pin2)
#define bittest1(D,i) ( ( D ) & ( ( ( unsigned long ) 1 << ( i ) ) ) ) && ( ( ( unsigned long ) 1 << ( i ) ) ) )
#define CLK2 LATCbits.LATC0 //PIN_D3 (pin41)
#define DATA_S2 LATCbits.LATC1 //PIN_D2 (pin40)
#define LE2 LATCbits.LATC2 //PIN_C2 (pin 36)
#define CE2 LATAbits.LATA5 //PIN_C1 (pin35)
#define bittest2(D,i) ( ( D ) & ( ( ( unsigned long ) 1 << ( i ) ) ) ) && ( ( ( unsigned long ) 1 << ( i ) ) ) )
static const unsigned long
↪ FREQ_INIT1[1][6]={0x9600000,0x8000011,0x1028E42,0x4B3,0xB0403C,0x580005}; //300MHz
static const unsigned long
↪ FREQ_INIT2[1][6]={0x9600020,0x80000C9,0x1028E42,0x4B3,0xB0403C,0x580005}; //300MHz+1kHz
```

### 9.5.4. Función de parpadeo de LEDs

```
void titilar(int led){
    if (led == 1) {
        LED1 = 1;
        for (int m = 0; m <= 1; m++) {
            __delay_ms(10);
        }
        LED1 = 0;
    }
    else if (led == 2) {
        LED2 = 1;
        for (int m = 0; m <= 1; m++) {
            __delay_ms(10);
        }
        LED2 = 0;
    }
}
```

### 9.5.5. Función de carga de registros

```

void set_freq_values_1(int initFreq){
    CE1 = 0;          //CE Select device
    for (int i = 0; i <= 5; i++)          // para 6 registros
    {
        LE1 = 0;
        for (int m = 0; m <= 31; m++)    //para 32 bits
        {
            CLK1 = 0;
            DATA_S1 = bittest1(FREC_1[initFreq][5-i],31-m);
            CLK1 = 1;
        }
        CLK1 = 0;
        DATA_S1 = 1;
        LE1 = 1;
    }
    CE1 = 1;          //CE Deselect device
    LE1 = 0;
    __delay_ms(1);
    LE1 = 1;
    titilar(1);
}

void set_freq_values_2(int initFreq){
    CE2 = 0;          //CE Select device
    for (int i = 0; i <= 5; i++)          // para 6 registros
    {
        LE2 = 0;
        for (int m = 0; m <= 31; m++)    //para 32 bits
        {
            CLK2 = 0;
            DATA_S2 = bittest2(FREC_2[initFreq][5-i],31-m);
            CLK2 = 1;
        }
        CLK2 = 0;
        DATA_S2 = 1;
        LE2 = 1;
    }
    CE2 = 1;          //CE Deselect device
    LE2 = 0;
    __delay_ms(1);
    LE2 = 1;
    titilar(2);
}

```

### 9.5.6. Funciones UART

```

void UART_Init(unsigned long baudrate) {
    unsigned int x;
    x = (_XTAL_FREQ - baudrate * 64) / (baudrate * 64); // Cálculo del baud rate para BRGH
    ↪ = 0
    if (x > 255) {
        x = (_XTAL_FREQ - baudrate * 16) / (baudrate * 16); // Cálculo del baud rate para
        ↪ BRGH = 1
        BRGH = 1; // Alta velocidad
    } else {
        BRGH = 0; // Baja velocidad
    }
    SPBRG = x; // Asignar valor calculado a SPBRG
}

```

```

    SYNC = 0;          // Modo asíncrono
    SPEN = 1;         // Habilitar puerto serie (RX/TX)
    TXEN = 1;         // Habilitar transmisión
    CREN = 1;         // Habilitar recepción
    TX9 = 0;          // Modo de 8 bits
    RX9 = 0;          // Modo de 8 bits
}

char UART_Read() {
    while (!RCIF);    // Esperar hasta que se reciba un dato
    return RCREG;     // Retornar el dato recibido
}

void UART_Write(char data) {
    while (!TXIF);    // Esperar hasta que el buffer esté vacío
    TXREG = data;     // Cargar el registro de transmisión con el dato
}

void UART_Write_32bits(unsigned long valor) {
    for (int i = 3; i >= 0; i--) { // Enviar de byte más significativo a menos
        ↪ significativo
        UART_Write((valor >> (i * 8)) & 0xFF); // Enviar byte por byte
    }
}

void UART_Write_Registers(char setID) {
    if (setID == 0x01) {
        // Transmitir el conjunto 1
        for (int i = 0; i < 6; i++) {
            UART_Write_32bits(FREC_1[0][i]);
        }
        // Transmitir el conjunto 2
        for (int i = 0; i < 6; i++) {
            UART_Write_32bits(FREC_2[0][i]);
        }
    }
}

void UART_Write_Text(char* text) {
    int i;
    for (i = 0; text[i] != '\0'; i++) {
        UART_Write(text[i]);
    }
}

```

### 9.5.7. Cálculo de Registros

```

void obtener_parametros(double f_out, int* div, int* frac, int* mod) {
    // Asignar FRAC y MOD fijos para todos los casos, ya que f_out es siempre entero
    *frac = 0;
    *mod = 2;
    // Asignar divisor basado en el rango de f_out
    if (f_out >= 300.0 && f_out <= 550.0) {
        *div = 8;
    } else if (f_out > 550.0 && f_out <= 1100.0) {
        *div = 4;
    } else if (f_out > 1100.0 && f_out <= 2200.0) {
        *div = 2;
    } else if (f_out > 2200.0 && f_out <= 3000.0) {

```

```

    *div = 1;
}
}

void obtener_parametros2(double f_out, int* div, int* frac, int* mod) {
    if (f_out >= 300.0 && f_out <= 550.0) {
        *div = 8;
        *mod = 125; // MOD estándar para este rango
        *frac = 2;
    } else if (f_out > 550.0 && f_out <= 1100.0) {
        *div = 4;
        *mod = 125; // MOD estándar para este rango
        *frac = 1;
    } else if (f_out > 1100.0 && f_out <= 2200.0) {
        *div = 2;
        *mod = 250; // MOD estándar para este rango
        *frac = 1;
    } else if (f_out > 2200.0 && f_out <= 3000.0) {
        *div = 1;
        *mod = 500; // MOD estándar para este rango
        *frac = 1;
    } else {
        return; // Salir si la frecuencia no está en el rango permitido
    }
}

int calcular_int(double f_out, double f_pfd, int div, int frac, int mod) {
    return (int)((f_out * div / f_pfd) - ((double)frac / (double)mod));
}

unsigned long calcular_registro1(int mod, int prescaler_8_9) {
    int control_bits = 0b001;
    int prescaler_bit = (prescaler_8_9) ? 1 : 0;
    int phase_adjust_bit = 0;
    int phase_value = 0x001;
    return ((unsigned long)phase_adjust_bit << 28) | ((unsigned long)prescaler_bit << 27) |
           ((unsigned long)phase_value << 15) | ((unsigned long)mod << 3) | control_bits;
}

unsigned long obtener_registro4(int div) {
    switch (div) {
        case 1: return 0x80403C;
        case 2: return 0x90403C;
        case 4: return 0xA0403C;
        case 8: return 0xB0403C;
        default: return 0; // En caso de error
    }
}

void cargar_freq1(double f_out) {
    double f_pfd = 0.5; // PFD en MHz
    int prescaler_8_9 = 1;
    int div, frac, mod;
    obtener_parametros(f_out, &div, &frac, &mod);
    int n_int = calcular_int(f_out, f_pfd, div, frac, mod);
    // Registro 0: INT y FRAC
    FREC_1[0][0] = ((unsigned long)n_int << 15) | ((unsigned long)frac << 3) | 0x0;
    // Registro 1: MOD, prescaler, fase
    FREC_1[0][1] = calcular_registro1(mod, prescaler_8_9);
}

```

```

    // Registro 4: Depende de DIV
    FREC_1[0][4] = obtener_registro4(div);
    set_freq_values_1(0); // Programar el sintetizador 1
}

void cargar_freq2(double f_out) {
    double f_pfd = 0.5; // PFD en MHz
    int prescaler_8_9 = 1;
    int div, frac, mod;
    obtener_parametros2(f_out, &div, &frac, &mod);
    int n_int = calcular_int(f_out, f_pfd, div, frac, mod);
    // Registro 0: INT y FRAC
    FREC_2[0][0] = ((unsigned long)n_int << 15) | ((unsigned long)frac << 3) | 0x0;
    // Registro 1: MOD, prescaler, fase
    FREC_2[0][1] = calcular_registro1(mod, prescaler_8_9);
    // Registro 4: Depende de DIV
    FREC_2[0][4] = obtener_registro4(div);
    set_freq_values_2(0); // Programar el sintetizador 2
}

```

### 9.5.8. Programa principal: Main

```

void set_registers(void);
void indicators(void);

void main (void) {
    UART_Init(9600); // Inicializar UART a 9600 baudios
    set_registers();
    indicators();
    while(1){
        char setID = UART_Read(); // Leer el ID del conjunto
        if (setID == 0x01) {
            // Leer dos bytes de UART para formar la frecuencia
            unsigned char freq_high = UART_Read(); // Leer el byte alto
            unsigned char freq_low = UART_Read(); // Leer el byte bajo
            // Combinar los dos bytes en un entero de 16 bits
            unsigned int freq = (freq_high << 8) | freq_low;
            // Verificar si la frecuencia está en el rango permitido (300 a 3000 MHz)
            // __delay_ms(10);
            // Llamar a las funciones con la frecuencia leída
            if (freq >= 300 && freq <= 3000) {
                double f_out = (double)freq; // Convertir a double para usar en
                ↪ cargar_freq1 y cargar_freq2
                // Llamar a las funciones con la frecuencia leída
                cargar_freq1(f_out);
                cargar_freq2(f_out);
            }
            for (int m = 0; m <= 5; m++) {
                __delay_ms(10);
            }
        }
        //UART_Write_Registers(setID);
    }
}

```

### 9.5.9. Funciones de inicio

```

void set_registers(void){
    for (int i = 0; i < 6; i++) {

```

```

        FREQ_1[0][i] = FREQ_INIT1[0][i];
    }
        for (int i = 0; i < 6; i++) {
            FREQ_2[0][i] = FREQ_INIT2[0][i];
        }
    /*Inicialización de variables*/
    SPPCON = 0 ; //disable SPP
    ADCON0 = 0 ; //disable ADC function
    CVRCON = 0 ; //more disables
    SSPCON1 = 0 ; //disable SPI functionality
    CCP1CON = 0 ; //disable both ccp modules
    ADCON1 = 0x0F ;
    //Para SPI
    SSPSTAT = 0b00000000; //0x00 De acuerdo a la hoja de datos
    SSPCON1 = 0b00110001; // De acuerdo a la hoja de datos
    //Seteo de los TRIS para el SPI del generador.
    //como salidas para enviar información.
    TRISDbits.TRISD7 = 0;
    TRISDbits.TRISD6 = 0;
    TRISDbits.TRISD5 = 0;
    TRISDbits.TRISD4 = 0;
    TRISCbits.TRISC0 = 0;
    TRISCbits.TRISC1 = 0;
    TRISCbits.TRISC2 = 0;
    TRISAbits.TRISA5 = 0;
    //Init Para SPI 1
    CE1 = 1;
    DATA_S1 = 1;
    LE1 = 1;
    CLK1 = 0;
    __delay_ms(10);
    set_frec_values_1(0);
    //Init Para SPI 2
    CE2 = 1;
    DATA_S2 = 1;
    LE2 = 1;
    CLK2 = 0;
    __delay_ms(10);
    set_frec_values_2(0);
}

void indicators(void){
    TRISDbits.TRISD0 = 0;
    TRISDbits.TRISD1 = 0;
    titilar(1);
    titilar(2);
}

```

## 9.6. Algoritmos principales en Python

### 9.6.1. Librerías

```
import sounddevice as sd
import numpy as np
from scipy import interpolate
from scipy.signal import butter, filtfilt, savgol_filter
from scipy.fft import fft, fftfreq
import time
import serial
import time
import plotly.io as pio
pio.renderers.default = "notebook"
import plotly.graph_objects as go
```

### 9.6.2. Envío de frecuencias por UART

```
def enviar_frecuencias_por_uart(frecuencia):
    global ser
    time.sleep(0.01)

    frecuencia = int(frecuencia)
    hex_str = hex(frecuencia)[2:].zfill(4)
    parte1 = hex_str[:2].upper()
    parte2 = hex_str[2:].upper()

    paquete1 = bytearray()
    paquete1.append(0x01)
    ser.write(paquete1)

    paquete1 = bytearray.fromhex(parte1)
    ser.write(paquete1)

    paquete2 = bytearray.fromhex(parte2)
    ser.write(paquete2)
```

### 9.6.3. Adquisición de señales

```
def capture_audio():
    input_device_index = 1 # Índice del dispositivo de entrada
    sd.default.device = (input_device_index, None) # (entrada, salida)

    audio_data = sd.rec(int(duration * sampling_rate), samplerate=sampling_rate, channels=2,
        ↪ dtype='float64')
    sd.wait()

    start_index = int(0.2 * duration * sampling_rate)
    # Descartar el 20% de las primeras muestras
    left_channel = audio_data[start_index:, 0] # Canal izquierdo
    right_channel = audio_data[start_index:, 1] # Canal derecho
    # Todas las muestras
    #left_channel = audio_data[:, 0] # Canal izquierdo
    #right_channel = audio_data[:, 1] # Canal derecho

    return left_channel, right_channel
```

### 9.6.4. Filtro digital pasabanda

```
def butter_bandpass(lowcut, highcut, fs, order=2):
    nyq = 0.5 * fs # Frecuencia de Nyquist
    low = lowcut / nyq
    high = highcut / nyq
    b, a = butter(order, [low, high], btype='band')
    return b, a

# Función para aplicar el filtro
def apply_bandpass_filter(data, lowcut, highcut, fs, order=2):
    b, a = butter_bandpass(lowcut, highcut, fs, order=order)
    y = filtfilt(b, a, data)
    return y
```

### 9.6.5. Cálculo de amplitud y fase mediante FFT

```
def fft_amplitude_phase(signal, sampling_rate, duration):
    # Calcular la FFT de la señal
    fft_signal = np.fft.fft(signal)
    frequencies = np.fft.fftfreq(len(signal), 1 / sampling_rate)
    N = duration * sampling_rate
    amps = np.array((2.0 / N) * np.abs(fft_signal))

    # Limitar las frecuencias a las positivas
    freqs_positive = frequencies[:int(N // 2)]
    amps_positive = amps[:int(N // 2)]
    # Definir la frecuencia objetivo y la tolerancia
    target_freq = 1000.0 # 1 kHz
    tolerance = 50.0 # Tolerancia de ±900 Hz
    # Encontrar los índices de las frecuencias cercanas a 1 kHz dentro de la tolerancia
    indices = np.where((freqs_positive >= (target_freq - tolerance)) & (freqs_positive <=
    ↪ (target_freq + tolerance)))[0]
    # Seleccionar la componente con la mayor amplitud
    max_index = indices[np.argmax(amps_positive[indices])]
    # Calcular la fase y amplitud de esa componente
    phase = -np.degrees(np.angle(fft_signal[max_index])) # en grados
    amp = amps_positive[max_index]

    return amp, phase

def phase_adjust(diff_phase):
    x = diff_phase
    if (x > 180.0):
        x = x - 360.0
    if (x < -180):
        x = x + 360.0
    return x

def calculate_amplitude_phase(signal_1, signal_2, sampling_rate, duration):
    amp_1, phase_1 = fft_amplitude_phase(signal_1, sampling_rate, duration)
    amp_2, phase_2 = fft_amplitude_phase(signal_2, sampling_rate, duration)
    dif_phase = phase_adjust(-phase_1+phase_2)
    transfer = amp_2/amp_1
    return transfer, dif_phase
```



### 9.6.6. Grabación y procesamiento de señales

```

def s_parameter(amp,phase):
    s = amp* np.exp(1j*(np.radians(phase)))
    return s

def s11_capture(duration,sampling_rate):
    # Capturar las señales de audio
    ix_signal, rev_signal = capture_audio()
    time_axis = np.linspace(0, duration, len(ix_signal))

    # Aplicar filtro pasa banda a ambas señales
    lowcut = 950.0
    highcut = 1050.0
    ix_signal_filtered = apply_bandpass_filter(ix_signal, lowcut, highcut, sampling_rate)
    rev_signal_filtered = apply_bandpass_filter(rev_signal, lowcut, highcut, sampling_rate)

    # Calcular Amplitud y Fase
    S11_amp, S11_phase =
    ↪ calculate_amplitude_phase(ix_signal_filtered,rev_signal_filtered,sampling_rate,duration)
    S11_amp_adjust = -S11_amp*286.0/40.0

    # Generar parámetro S:
    sp = s_parameter(S11_amp_adjust,S11_phase)
    return sp

def s21_capture(duration,sampling_rate):
    # Capturar las señales de audio
    tx_signal, ix_signal = capture_audio()
    time_axis = np.linspace(0, duration, len(ix_signal))

    # Aplicar filtro pasa banda a ambas señales
    lowcut = 950.0
    highcut = 1050.0
    ix_signal_filtered = apply_bandpass_filter(ix_signal, lowcut, highcut, sampling_rate)
    tx_signal_filtered = apply_bandpass_filter(tx_signal, lowcut, highcut, sampling_rate)

    # Calcular Amplitud y Fase
    S21_amp, S21_phase =
    ↪ calculate_amplitude_phase(ix_signal_filtered,tx_signal_filtered,sampling_rate,duration)
    S21_amp_adjust = S21_amp*25.0
    # Generar parámetro S:
    sp = s_parameter(S21_amp_adjust,S21_phase)
    return sp

```

### 9.6.7. Cálculo de Registros

```

def obtener_parametros(f_out):
    if 300 <= f_out < 550:
        div = 8
        if f_out.is_integer():
            frac = 0
            mod = 2
        else:
            frac = 2
            mod = 125
    elif 550 <= f_out < 1100:
        div = 4
        if f_out.is_integer():

```

```

        frac = 0
        mod = 2
    else:
        frac = 1
        mod = 125
elif 1100 <= f_out < 2200:
    div = 2
    if f_out.is_integer():
        frac = 0
        mod = 2
    else:
        frac = 1
        mod = 250
elif 2200 <= f_out <= 3000:
    div = 1
    if f_out.is_integer():
        frac = 0
        mod = 2
    else:
        frac = 1
        mod = 500
else:
    raise ValueError("Fout fuera del rango permitido")

return div, frac, mod

# Función para calcular el valor de INT basado en la ecuación
def calcular_int(f_out, f_pfd, div, frac, mod):
    # Usamos la ecuación Fout = (INT + FRAC / MOD) * (PDF / DIV)
    int_valor = (f_out * div / f_pfd) - (frac / mod)
    return int(int_valor)

# Función para determinar el valor del Registro 1 según MOD
def calcular_registro1(mod, prescaler_8_9):
    # Bits de control fijos
    control_bits = 0b001
    # Prescaler: 1 para 8/9, 0 para 4/5
    prescaler_bit = 1 if prescaler_8_9 else 0
    # Phase Adjust (por defecto lo mantendremos en 0)
    phase_adjust_bit = 0
    # Phase Value (utilizamos el valor recomendado que es 0)
    phase_value = 0x001 # 12 bits para el valor de fase
    # Construir el registro 1
    registro1 = (phase_adjust_bit << 28) | (prescaler_bit << 27) | (phase_value << 15) |
    ↪ (mod << 3) | control_bits
    return registro1

# Función para determinar el valor del Registro 4 según DIV
def obtener_registro4(div):
    if div == 1:
        return 0x80403C
    elif div == 2:
        return 0x90403C
    elif div == 4:
        return 0xA0403C
    elif div == 8:
        return 0xB0403C
    else:
        raise ValueError("DIV no válido")

```

```

# Función principal para calcular los registros del ADF4351
def calcular_registros(f_out):
    f_pfd = 0.5 # PDF en MHz
    prescaler_8_9 = True # Definir si se usa prescaler 8/9
    # Obtener los valores de DIV, FRAC y MOD
    div, frac, mod = obtener_parametros(f_out)
    # Calcular el valor de INT
    n_int = calcular_int(f_out, f_pfd, div, frac, mod)
    # Registro 0: Contiene INT (N) y FRAC
    r0 = (n_int << 15) | (frac << 3) | 0x0 # Registro 0: Bits 31-15 para N, 14-3 para FRAC,
    ↪ bits 2-0 en 0
    # Registro 1: Contiene MOD y bits de control, preescaler, fase
    r1 = calcular_registro1(mod, prescaler_8_9)
    # Registro 2: Configuración de prescaler y otros controles
    prescaler_bit = 1 if prescaler_8_9 else 0
    r2 = (prescaler_bit << 28) | 0x28E42 # Ajuste para incluir prescaler
    # Registro 3: Control de cargas, feedback, y fase de sincronización
    r3 = 0x004B3 # Configuración estándar
    # Registro 4: Depende de DIV
    r4 = obtener_registro4(div) # Determina el valor de R4 basado en DIV
    # Registro 5: Configuración fija para el ADF4351 (modo de trabajo)
    r5 = 0x580005 # Configuración estándar
    return [r0, r1, r2, r3, r4, r5]

```

### 9.6.8. Función de barrido en frecuencia

```

def medicion_s(tipo,duration,sampling_rate):
    s = []
    if tipo == "S11":
        s = s11_capture(duration,sampling_rate)
    elif tipo == "S21":
        s = s21_capture(duration,sampling_rate)
    return s

# Barrido de frecuencias
def barrido_frecuencia(f_in, deltaf, f_end,tipo,duration,sampling_rate,muestras):
    global ser
    n = int(1+(f_end-f_in)/deltaf)
    SS = np.empty(n, dtype=complex)
    sum = np.zeros(n, dtype=complex)
    freqs = np.empty(n)
    j = 0
    f_out = f_in
    print(f"Measuring... from {f_in} MHz to {f_end} MHz")
    try:
        for k in range(muestras):
            j = 0
            f_out = f_in # Reiniciar f_out en cada iteración de las muestras
            while f_out <= f_end:
                enviar_frecuencias_por_uart(f_out)
                time.sleep(0.02)
                # Guardar la medición y la frecuencia correspondiente
                SS[j] = medicion_s(tipo, duration, sampling_rate)
                sum[j] = sum[j] + SS[j]
                freqs[j] = f_out
                f_out += deltaf
                j += 1 # Incrementar el índice de muestras válidas

```

```

except Exception as e:
    print(f"ERROR: {e}")
finally:
    ser.close()
    print("Fin.")

# Retornar las frecuencias y el promedio de las muestras
return freqs[:,j], (sum[:,j] / muestras)

```

### 9.6.9. Error en reflexión y correctores

```

def error_calculator(GM_CC,GM_50,GM_CA,n):
    e_00 = np.empty(n, dtype=complex)
    e_11 = np.empty(n, dtype=complex)
    delta_e = np.empty(n, dtype=complex)

    for i in range(n):
        A = np.array([
            [1.0,-GM_CC[i],1.0], #CC
            [1.0,0.0,0.0],      #50
            [1.0,GM_CA[i],-1.0] #CA
        ],dtype=complex)
        B = np.array([GM_CC[i],GM_50[i],GM_CA[i]])
        sol = np.linalg.solve(A, B)
        #sol = np.float64(sol)
        e_00[i] = sol[0]
        e_11[i] = sol[1]
        delta_e[i] = sol[2]

    return e_00,e_11,delta_e

# Gamma corrector
def gamma_corrector(GM,GM_CC,GM_50,GM_CA,n):
    e_00,e_11,delta_e = error_calculator(GM_CC,GM_50,GM_CA,n)
    Gamma = np.empty(n, dtype=complex)
    for i in range(n):
        Gamma[i] = (GM[i]-e_00[i])/(GM[i]*e_11[i]-delta_e[i])
    return Gamma

# Transfer corrector
def transfer_corrector(GDUT,e30,e1032,n):
    G = np.empty(n, dtype=complex)
    for i in range(n):
        G[i] = (GDUT[i]-e30[i])/e1032[i]
    return G

```

### 9.6.10. Funciones gráficas

```

def graficar_magnitud_y_fase_vs_frecuencia(S21_freqs, S21_Att):
    plt.figure(figsize=(10, 6), facecolor='white')
    ax = plt.gca()
    ax.set_facecolor('white')

    # Subplot para la magnitud
    plt.subplot(2, 1, 1) # 2 filas, 1 columna, primera posición
    # Calcular la magnitud y convertirla a dB
    magnitud = np.abs(S21_Att)
    magnitud_dB = 20.0 * np.log10(magnitud)
    plt.plot(S21_freqs, magnitud_dB, marker='.', linewidth=2.5)

```

```

plt.title('Magnitud y Fase')
plt.xlabel('Frecuencia (MHz)')
plt.ylabel('Magnitud (dB)')
plt.legend()
plt.grid(True)

# Subplot para la fase
plt.subplot(2, 1, 2) # 2 filas, 1 columna, segunda posición
# Calcular la fase en grados
fase = np.angle(S21_Att, deg=True)
plt.plot(S21_freqs, fase, marker='.', linewidth=2.5)
plt.xlabel('Frecuencia (MHz)')
plt.ylabel('Fase (grados)')
plt.grid(True)
# Ajustar el espaciado para que no se superpongan los subplots
plt.tight_layout()
# plt.savefig("Atenuador_uVNA.png", dpi=300, bbox_inches='tight', format='png') #
# ↪ cambiar formato a 'pdf' o 'svg'
plt.show()

```

*# Gráfico de Smith*

```

def plot_smith(ZL,Z0=50.0):
    Zn = ZL/Z0
    Imags = (Zn.imag)
    Reals = (Zn.real)
    fig = go.Figure(go.Scattersmith(imag= Imags,real = Reals))
    #fig.write_html("output.html")
    fig.show(renderer="browser")

```

### 9.6.11. Mediciones S21/S12

```

sampling_rate = 44100 # Frecuencia de muestreo
duration = 0.05 # Duración de la grabación en segundos
sd.default.device = (input_device_index, None) # (entrada, salida)
puerto_uart = 'COM14'

# Frequency Range
f_in = 300 #MHz
deltaf = 100 #MHz
f_end = 3000 #MHz
muestras = 1
tipo = "S21"

# Leakage Error
ser = serial.Serial(puerto_uart, baudrate=9600, timeout=1) # Aumenta el timeout si es
# ↪ necesario
_,e30 = barrido_frecuencia(f_in, deltaf, f_end,tipo,duration,sampling_rate,muestras)

# Cable THR Error
ser = serial.Serial(puerto_uart, baudrate=9600, timeout=1) # Aumenta el timeout si es
# ↪ necesario
_,e1032 = barrido_frecuencia(f_in, deltaf, f_end,tipo,duration,sampling_rate,muestras)

# DUT TRANSFER
ser = serial.Serial(puerto_uart, baudrate=9600, timeout=1) # Aumenta el timeout si es
# ↪ necesario
freqs,G21_sin = barrido_frecuencia(f_in, deltaf, f_end,tipo,duration,sampling_rate,muestras)
points = len(G21_sin)
S21 = transfer_corrector(G21_sin,e30,e1032,points)

```

```
graficar_magnitud_y_fase_vs_frecuencia(freqs, S21)
```

### 9.6.12. Mediciones S11/S22

```
def impedancimeter(Gamma,Zo = 50.0):
    n = len(Gamma)
    ZL = np.empty(n, dtype=complex)
    for i in range(n):
        ZL[i] = Zo * (1 + Gamma[i])/(1 - Gamma[i])
    return ZL

sampling_rate = 44100 # Frecuencia de muestreo
duration = 0.05 # Duración de la grabación en segundos
sd.default.device = (input_device_index, None) # (entrada, salida)
puerto_uart = 'COM14'

# Frequency Range
f_in = 300.00 #MHz
deltaf = 100.00 #MHz
f_end = 3000.00 #MHz
muestras = 5
tipo = "S11"

# Carga Corto Circuito
ser = serial.Serial(puerto_uart, baudrate=9600, timeout=1) # Aumenta el timeout si es
↳ necesario
_,GM_CC = barrido_frecuencia(f_in, deltax, f_end,tipo,duration,sampling_rate,muestras)

# Carga 50 Ohm
ser = serial.Serial(puerto_uart, baudrate=9600, timeout=1) # Aumenta el timeout si es
↳ necesario
_,GM_50 = barrido_frecuencia(f_in, deltax, f_end,tipo,duration,sampling_rate,muestras)

# Carga Circuito Abierto
ser = serial.Serial(puerto_uart, baudrate=9600, timeout=1) # Aumenta el timeout si es
↳ necesario
_,GM_CA = barrido_frecuencia(f_in, deltax, f_end,tipo,duration,sampling_rate,muestras)

# DUT REFLECTION
ser = serial.Serial(puerto_uart, baudrate=9600, timeout=1) # Aumenta el timeout si es
↳ necesario
freqs,GM_ZL = barrido_frecuencia(f_in, deltax, f_end,tipo,duration,sampling_rate,muestras)
points = len(GM_ZL)
Gamma = gamma_corrector(GM_ZL,GM_CC,GM_50,GM_CA,points)
graficar_magnitud_y_fase_vs_frecuencia(freqs, Gamma)
ZL = impedancimeter(Gamma)
plot_smith(ZL)
```

### 9.6.13. Función de Suavizado Savitsky-Golay

```
def smoothing_savgol2(s):
    amplitude = np.abs(s)
    #phase = np.angle(s)

    amplitude_smooth = savgol_filter(amplitude, window_length=17, polyorder=3)
    #phase_smooth = savgol_filter(phase, window_length=9, polyorder=3)
```

```
S_real_smooth = savgol_filter(s.real, window_length=17, polyorder=3)
S_imag_smooth = savgol_filter(s.imag, window_length=17, polyorder=3)
phase_smooth = np.angle(S_real_smooth + 1j * S_imag_smooth)

S_smooth = amplitude_smooth * np.exp(1j * phase_smooth)
return S_smooth
```

## Referencias

- [1] Inc. Tektronix. *Vector Network Analysis: A Practical Introduction*. Inf. téc. Tektronix, 2023. URL: [https://download.tek.com/document/70W\\_60918\\_0\\_Tek\\_VNA\\_PR.pdf](https://download.tek.com/document/70W_60918_0_Tek_VNA_PR.pdf).
- [2] Rigol Technologies, Inc. *Rigol RSA3030N 3GHz Real-Time Spectrum Analyser*. URL: <https://www.rigol-uk.co.uk/product/rigol-rsa3030n-3ghz-real-time-spectrum-analyser/>.
- [3] Microwaves101. *Coupler Fundamentals*. 2024. URL: <https://www.microwaves101.com/encyclopedias/coupler-fundamentals>.
- [4] David M. Pozar. *Microwave Engineering*. 4th. Wiley, 2012.
- [5] Analog Devices, Inc. *ADL5920: 9 kHz to 7 GHz Directional Bridge and Dual RMS Power Detector*. URL: <https://www.analog.com/en/products/adl5920.html>.
- [6] W. B. Jones. *High Performance Antenna Matching Networks*. Inf. téc. American Radio Relay League, 2007. URL: [https://w5big.com/QST\\_Article.pdf](https://w5big.com/QST_Article.pdf).
- [7] Altium, Inc. *Altium Designer*. 2024. URL: <https://www.altium.com/es/altium-designer>.
- [8] Analog Devices, Inc. *LTspice Simulator*. 2024. URL: <https://www.analog.com/en/resources/design-tools-and-calculators/ltspice-simulator.html>.
- [9] Dassault Systèmes. *CST Studio Suite*. 2024. URL: <https://www.3ds.com/products/simulia/cst-studio-suite>.
- [10] QucsStudio. *QucsStudio*. 2024. URL: <https://quesstudio.de/>.
- [11] Microchip Technology Inc. *PIC18F4550*. URL: <https://www.microchip.com/en-us/product/pic18f4550>.
- [12] Texas Instruments. *MAX232: Dual EIA-232 Drivers/Receivers*. Inf. téc. URL: [https://www.ti.com/lit/ds/symlink/max232.pdf?ts=1731395987713&ref\\_url=https%253A%252F%252Fwww.google.com%252F](https://www.ti.com/lit/ds/symlink/max232.pdf?ts=1731395987713&ref_url=https%253A%252F%252Fwww.google.com%252F).
- [13] ProtoExpress. *High-Speed PCB Design Guide*. ProtoExpress, 2024. URL: <https://s3vi.ndc.nasa.gov/ssri-kb/static/resources/High-Speed%20PCB%20Design%20Guide.pdf>.
- [14] Microchip Technology Inc. *MPLAB X IDE*. 2024. URL: <https://www.microchip.com/en-us/tools-resources/develop/mplab-x-ide>.
- [15] Microchip Technology Inc. *PICKit*. 2024. URL: <https://www.microchip.com/en-us/development-tool/pg164120>.
- [16] Analog Devices. *ADF4351: Wideband Synthesizer with Integrated VCO*. Inf. téc. URL: <https://www.analog.com/media/en/technical-documentation/data-sheets/ADF4351.pdf>.
- [17] ON Semiconductor. *2N3904: NPN Transistor*. Inf. téc. URL: <https://www.onsemi.com/download/data-sheet/pdf/2n3903-d.pdf>.
- [18] Altium. *Via Stitching and Via Shielding in PCB Design*. URL: <https://www.altium.com/documentation/altium-designer/via-stitching-via-shielding-pcb?srsltid=AfmBOoo7b3e1wgInixLixqW5H>.
- [19] NASA Glenn Research Center. *Tuned Wheatstone Bridge*. URL: <https://www.grc.nasa.gov/www/k-12/airplane/tunwheat.html>.
- [20] Analog Devices. «Bidirectional Bridge with Dual RMS Detectors Targets RF Power and Return Loss Measurement». En: *Microwave and RF* (). URL: <https://www.mwrf.com/technologies/components/article/21849864/analog-devices-bidirectional-bridge-with-dual-rms-detectors-targets-rf-power-return-loss-measurement>.



- [21] Robert Weber. *Introduction to Microwave Circuits*. Prentice Hall, 1994.
- [22] Hewlett-Packard. *Application Note S013: IAM-8 Series Active Mixer*. 1998. URL: [http://www.hp.woodshot.com/hprfhel/4\\_downld/lit/iclit/ans013.pdf](http://www.hp.woodshot.com/hprfhel/4_downld/lit/iclit/ans013.pdf).
- [23] Mini-Circuits. «Application Note AN10-006: Power Splitters». En: (2015). URL: [https://www.minicircuits.com/app/AN10-006.pdf?srsltid=AfmBOorHN7Q9w9Bkq1d7QK\\_KE8hQ28TkINLP3o-ts850NRXNpHyC661y](https://www.minicircuits.com/app/AN10-006.pdf?srsltid=AfmBOorHN7Q9w9Bkq1d7QK_KE8hQ28TkINLP3o-ts850NRXNpHyC661y).
- [24] Texas Instruments. «PCM2904 USB Audio Codec Data Sheet». En: (2007). URL: [https://www.ti.com/lit/ds/symlink/pcm2904.pdf?ts=1731479840906&ref\\_url=https%253A%252F%252Fwww.google.com%252F](https://www.ti.com/lit/ds/symlink/pcm2904.pdf?ts=1731479840906&ref_url=https%253A%252F%252Fwww.google.com%252F).
- [25] Microchip Technology Inc. *MCP6030 Operational Amplifier Data Sheet*. 2017. URL: <https://ww1.microchip.com/downloads/en/DeviceDoc/20001685E.pdf>.
- [26] Python Software Foundation. *Python Programming Language*. 2024. URL: <https://www.python.org/>.
- [27] Microsoft. *Visual Studio Code*. 2024. URL: <https://code.visualstudio.com/>.
- [28] Jupyter Project. *Project Jupyter*. 2024. URL: <https://jupyter.org/>.
- [29] Riverbank Computing Limited. *PyQt*. 2024. URL: <https://www.riverbankcomputing.com/software/pyqt/intro>.
- [30] Alan V. Oppenheim y Ronald W. Schaffer. *Discrete-Time Signal Processing*. 3rd. Prentice Hall, 2010.
- [31] Agilent Technologies. *NA Error Models and Cal Methods*. URL: [https://www.rfmentor.com/sites/default/files/NA\\_Error\\_Models\\_and\\_Cal\\_Methods.pdf](https://www.rfmentor.com/sites/default/files/NA_Error_Models_and_Cal_Methods.pdf).
- [32] Kenwood Corporation. *Kenwood PA-36 3A Power Supply*. URL: <https://www.manualslib.com/products/Kenwood-Pa36-3-4017308.html>.
- [33] Tektronix Inc. *TDS 210 / TDS 220 Oscilloscope Fact Sheet*. URL: <https://www.tek.com/en/documents/fact-sheet/tds-210-tds-220-fact-sheet>.
- [34] Keysight Technologies. *33120A Function / Arbitrary Waveform Generator (15 MHz)*. URL: <https://www.keysight.com/us/en/product/33120A/function--arbitrary-waveform-generator-15-mhz.html>.
- [35] Keysight Technologies. *8496B Manual Attenuator (18 GHz, 110 dB, 10 dB Steps)*. URL: <https://www.keysight.com/us/en/product/8496B/manual-attenuator-18-ghz-110-db-10-db-steps.html>.
- [36] UNIT. *UNIT 55 Multimeter*. URL: <https://elaso.com.ua/uploads/docs/-t4qqKaqDB.pdf>.
- [37] Pythoneers. «Introduction to the Savitzky-Golay Filter: A Comprehensive Guide Using Python». En: *Medium* (2020). URL: <https://medium.com/pythoneers/introduction-to-the-savitzky-golay-filter-a-comprehensive-guide-using-python-b2dd07a8e2ce>.
- [38] Bruce R. Archambeault. *EMC Control in PCB Design for Working Engineers*. IBM Corporation, 2010.
- [39] Viasion. *Rogers PCB vs FR4 PCB: Features, Advantages, and Applications*. URL: <https://www.viasion.com/es/blog/rogers-pcb-vs-fr4-pcb-features-advantages-and-applications/>.