

Herramientas y Recursos para el Desarrollo

Sistema Integral para Proveedores de Internet (ISP)

1. Entorno de Desarrollo

1.1 Herramientas de Desarrollo Generales

Herramienta	Propósito	Enlace
Git	Control de versiones	git-scm.com
Visual Studio Code	IDE principal	code.visualstudio.com
Docker	Contenerización	docker.com
Docker Compose	Orquestación de contenedores	docs.docker.com/compose
Postman	Pruebas de API	postman.com
DBeaver	Cliente de base de datos	dbeaver.io

1.2 Extensiones Recomendadas para VS Code

- ESLint: Linting para JavaScript
- Vetur/Volar: Soporte para Vue.js
- Docker: Integración con Docker
- PostgreSQL: Soporte para PostgreSQL
- REST Client: Cliente REST integrado
- Live Share: Colaboración en tiempo real

2. Stack Tecnológico

2.1 Frontend

Tecnología	Versión	Propósito	Enlace
Vue.js	3.x	Framework de interfaz de usuario	vuejs.org
Vuetify	3.x	Componentes Material Design	vuetifyjs.com
Axios	1.x	Cliente HTTP	axios-http.com
Vuex	4.x	Gestión de estado	vuex.vuejs.org
Vue Router	4.x	Enrutamiento	router.vuejs.org
Chart.js	3.x	Gráficos y visualizaciones	chartjs.org
Leaflet	1.x	Mapas interactivos	leafletjs.com

2.2 Backend

Tecnología	Versión	Propósito	Enlace
Node.js	18.x LTS	Entorno de ejecución	nodejs.org
Express	4.x	Framework web	expressjs.com
Sequelize	6.x	ORM para PostgreSQL	sequelize.org
InfluxDB Client	2.x	Cliente para InfluxDB	influxdata.com
Redis Client	4.x	Cliente para Redis	redis.js.org
Passport	0.6.x	Autenticación	passportjs.org
JWT	9.x	Tokens de autenticación	jwt.io
Multer	1.x	Carga de archivos	npmjs.com/package/multer

2.3 Bases de Datos

Tecnología	Versión	Propósito	Enlace
PostgreSQL	14.x	Base de datos principal	postgresql.org
InfluxDB	2.x	Base de datos de series temporales	influxdata.com
Redis	7.x	Caché y sesiones	redis.io

2.4 Infraestructura

Tecnología	Versión	Propósito	Enlace
Nginx	Última	Proxy inverso	nginx.com
Let's Encrypt	-	Certificados SSL	letsencrypt.org
Ubuntu Server	22.04 LTS	Sistema operativo	ubuntu.com/server

3. Bibliotecas y Clientes API

3.1 Integración con Equipos de Red

Biblioteca	Propósito	Enlace
node-routeros-api	Cliente API para Mikrotik	npmjs.com/package/node-routeros-api
ubnt-unms	Cliente para UNMS/UISP	npmjs.com/package/ubnt-unms
snmp-native	Monitoreo SNMP	npmjs.com/package/snmp-native

3.2 Integración con Servicios de Comunicación

Biblioteca	Propósito	Enlace
Nodemailer	Envío de correos	nodemailer.com
Telegraf	Cliente API para Telegram	telegraf.js.org
whatsapp-web.js	Cliente no oficial para WhatsApp	npmjs.com/package/whatsapp-web.js

3.3 Integración con Jellyfin

Biblioteca	Propósito	Enlace
jellyfin-apiclient	Cliente para Jellyfin	github.com/jellyfin/jellyfin-apiclient-javascript
better-sqlite3	Acceso a DB de JFA-GO	npmjs.com/package/better-sqlite3

4. Recursos y Documentación

4.1 Documentación Oficial de APIs

API	Documentación
Mikrotik RouterOS API	wiki.mikrotik.com/wiki/Manual
Ubiquiti UNMS/UISP API	help.ui.com/hc/en-us/articles/115000190833-UISP-API
Jellyfin API	api.jellyfin.org
Telegram Bot API	core.telegram.org/bots/api

4.2 Tutoriales y Guías

Recurso	Tema	Enlace
Vue.js	Guía oficial	vuejs.org/guide
Express.js	Getting Started	expressjs.com/starter/installing.html
Docker	Get Started	docs.docker.com/get-started
Digital Ocean	Configuración de Nginx	digitalocean.com/community/tutorials/how-to-install-nginx-on-ubuntu-22-04

4.3 Comunidades y Foros

Comunidad	Tema	Enlace
Stack Overflow	Programación general	stackoverflow.com
Reddit r/WISP	Proveedores de Internet inalámbricos	reddit.com/r/WISP
MikroTik Forum	Soporte Mikrotik	forum.mikrotik.com
UI Community	Soporte Ubiquiti	community.ui.com

5. Plantillas y Componentes

5.1 Plantillas de UI

Recurso	Tipo	Enlace
Vuetify UI Kit	Plantillas de componentes	vuetifyjs.com/en/getting-started/wireframes
Vue Admin Template	Dashboard administrativo	github.com/PanJiaChen/vue-admin-template
Tailwind UI	Componentes HTML/CSS	tailwindui.com

5.2 Iconos y Recursos Visuales

Recurso	Tipo	Enlace
Material Design Icons	Iconos	materialdesignicons.com
Font Awesome	Iconos	fontawesome.com
Unsplash	Imágenes libres	unsplash.com
unDraw	Ilustraciones SVG	undraw.co

6. Bibliotecas Especializadas

6.1 Monitoreo y Visualización

Biblioteca	Propósito	Enlace
ApexCharts	Gráficos avanzados	apexcharts.com
D3.js	Visualización de datos	d3js.org
vis-network	Visualización de topología de red	visjs.github.io/vis-network

6.2 Generación de Informes

Biblioteca	Propósito	Enlace
PDFKit	Generación de PDF	pdfkit.org
ExcelJS	Generación de Excel	github.com/exceljs/exceljs
html-to-pdfmake	Conversión HTML a PDF	npmjs.com/package/html-to-pdfmake

7. Recursos para Desarrollo Acelerado

7.1 Generadores de Código

Herramienta	Propósito	Enlace
Sequelize CLI	Generación de modelos y migraciones	npmjs.com/package/sequelize-cli
Vue CLI	Creación de proyectos Vue	cli.vuejs.org
Swagger Codegen	Generación de clientes API	swagger.io/tools/swagger-codegen

7.2 Plantillas de Backend

Recurso	Propósito	Enlace
Express Generator	Estructura básica Express	expressjs.com/starter/generator.html
Node.js Best Practices	Guías de implementación	github.com/goldbergonyi/nodebestpractices

8. Requisitos de Desarrollo

8.1 Hardware Recomendado

Componente	Requisito Mínimo	Recomendado
CPU	4 núcleos	8 núcleos
RAM	8 GB	16 GB
Almacenamiento	SSD 256 GB	SSD 512 GB
Conexión	50 Mbps	100+ Mbps

8.2 Software Local

Software	Propósito	Plataformas
Docker Desktop	Desarrollo local con Docker	Windows, macOS
Node.js	Entorno de desarrollo JavaScript	Windows, macOS, Linux
Git	Control de versiones	Windows, macOS, Linux
VS Code	Editor de código	Windows, macOS, Linux

9. Recursos de Aprendizaje

9.1 Cursos Recomendados

Curso	Tema	Plataforma
Vue.js - The Complete Guide	Vue.js	Udemy
Node.js: The Complete Guide	Node.js	Udemy
Docker Mastery	Docker	Udemy
SQL and PostgreSQL: The Complete Developer's Guide	PostgreSQL	Udemy

9.2 Libros Recomendados

Libro	Tema	Autor
Node.js Design Patterns	Node.js	Mario Casciaro
Full-Stack Vue.js 2 and Laravel 5	Vue.js	Anthony Gore
Docker in Action	Docker	Jeff Nickoloff
Database Design for Mere Mortals	Diseño de BD	Michael J. Hernandez

10. Soluciones de Monitoreo para Producción

10.1 Herramientas de Logging

Herramienta	Propósito	Enlace
Winston	Logging para Node.js	npmjs.com/package/winston
Morgan	Logging HTTP	npmjs.com/package/morgan
ELK Stack	Análisis de logs	elastic.co/elastic-stack

10.2 Monitoreo de Aplicaciones

Herramienta	Propósito	Enlace
PM2	Gestor de procesos Node.js	pm2.keymetrics.io
Prometheus	Monitoreo de métricas	prometheus.io
Grafana	Visualización de métricas	grafana.com
Sentry	Monitoreo de errores	sentry.io

11. Recursos para Seguridad

11.1 Herramientas de Seguridad

Herramienta	Propósito	Enlace
Helmet	Seguridad para Express	npmjs.com/package/helmet
bcrypt	Hash de contraseñas	npmjs.com/package/bcrypt
OWASP ZAP	Análisis de vulnerabilidades	zapproxy.org
ESLint Security	Análisis estático de seguridad	npmjs.com/package/eslint-plugin-security

11.2 Guías de Seguridad

Recurso	Tema	Enlace
OWASP Top 10	Vulnerabilidades web comunes	owasp.org/Top10
NodeJS Security Cheat Sheet	Seguridad en Node.js	cheatsheetseries.owasp.org/cheatsheets/Nodejs_Security_Cheat_Sheet.html
Vue Security Guide	Seguridad en Vue.js	vuejs.org/guide/best-practices/security.html

12. Configuración de Entorno de Desarrollo Local

A continuación se detallan los pasos para configurar un entorno de desarrollo local completo para el Sistema ISP:

12.1 Instalación de Software Base

```
bash
```

```
# Instalación de Node.js (mediante NVM para mejor gestión de versiones)
```

```
curl -o- https://raw.githubusercontent.com/nvm-sh/nvm/v0.39.3/install.sh | bash
```

```
nvm install 18
```

```
nvm use 18
```

```
# Instalación de Docker y Docker Compose
```

```
# Para Ubuntu:
```

```
sudo apt update
```

```
sudo apt install -y docker.io docker-compose
```

```
sudo usermod -aG docker $USER
```

```
# Para Windows/macOS: Descargar Docker Desktop desde docker.com
```

12.2 Configuración del Proyecto

```
bash

# Clonar repositorio (o crear nuevo)
git clone <url-repositorio> isp-system
cd isp-system

# Configuración de estructura de directorios
mkdir -p backend/src/{controllers,models,routes,services,utils,middleware}
mkdir -p frontend/src/{components,views,store,router,assets,utils}
mkdir -p scripts docs

# Inicializar backend
cd backend
npm init -y
npm install express sequelize pg influx redis jsonwebtoken bcrypt cors helmet winston morgan dc
npm install --save-dev nodemon eslint eslint-plugin-security

# Inicializar frontend
cd ../frontend
npm init vue@latest # Y seguir las instrucciones del asistente
cd vue-project # O el nombre que hayas elegido
npm install axios vuex@next vue-router@next vuetify@next chart.js leaflet

# Volver a la raíz del proyecto
cd ../../
```

12.3 Inicialización de Docker Compose

Crear archivo `docker-compose.yml` en la raíz del proyecto:

yaml

version: '3.8'

services:

postgres:

image: postgres:14

container_name: isp_postgres

volumes:

- postgres_data:/var/lib/postgresql/data

environment:

POSTGRES_PASSWORD: postgres

POSTGRES_USER: postgres

POSTGRES_DB: isp_db

ports:

- "5432:5432"

restart: unless-stopped

influxdb:

image: influxdb:2.6

container_name: isp_influxdb

volumes:

- influxdb_data:/var/lib/influxdb2

environment:

- DOCKER_INFLUXDB_INIT_MODE=setup
- DOCKER_INFLUXDB_INIT_USERNAME=admin
- DOCKER_INFLUXDB_INIT_PASSWORD=password
- DOCKER_INFLUXDB_INIT_ORG=isporg
- DOCKER_INFLUXDB_INIT_BUCKET=network_metrics

ports:

- "8086:8086"

restart: unless-stopped

redis:

image: redis:7-alpine

container_name: isp_redis

volumes:

- redis_data:/data

ports:

- "6379:6379"

restart: unless-stopped

volumes:

postgres_data:

influxdb_data:

redis_data:

12.4 Configuración de Scripts de Desarrollo

Añadir los siguientes scripts al `package.json` del backend:

```
json

"scripts": {
  "start": "node src/index.js",
  "dev": "nodemon src/index.js",
  "lint": "eslint src/",
  "test": "jest"
}
```

Añadir los siguientes scripts al `package.json` del frontend:

```
json

"scripts": {
  "serve": "vue-cli-service serve",
  "build": "vue-cli-service build",
  "lint": "vue-cli-service lint"
}
```

13. Recursos de Mikrotik y Ubiquiti

13.1 Herramientas Útiles para Mikrotik

Herramienta	Propósito	Enlace
RouterOS API Client Generator	Generación de clientes API	github.com/aluisiora/routeros-api-php
The Dude	Monitoreo de red Mikrotik	mikrotik.com/thedude
WinBox	Cliente de administración	mikrotik.com/download

13.2 Recursos para Ubiquiti

Recurso	Propósito	Enlace
UNMS/UISP	Gestión unificada	ui.com/consoles/uisp
AirControl	Control de dispositivos	ui.com/download/utilities
Ubiquiti Device Discovery Tool	Descubrimiento de dispositivos	ui.com/download/utilities

14. Ejemplos de Código para Integraciones Clave

14.1 Conexión a Mikrotik RouterOS API

javascript

// Ejemplo básico de conexión a Mikrotik RouterOS

```
const RouterOSAPI = require('node-routeros-api');
```

// Crear conexión

```
const api = new RouterOSAPI({  
  host: '192.168.88.1',  
  user: 'admin',  
  password: 'password',  
  port: 8728  
});
```

// Conectar y ejecutar comandos

```
api.connect()  
  .then(() => {  
    console.log('Conectado a RouterOS');  
  
    // Obtener lista de interfaces  
    return api.write('/interface/print');  
  })  
  .then((interfaces) => {  
    console.log('Interfaces:', interfaces);  
  
    // Cerrar conexión  
    api.close();  
  })  
  .catch((err) => {  
    console.error('Error de conexión:', err);  
  });
```

14.2 Integración con API de Jellyfin

javascript

// Ejemplo básico de conexión a API de Jellyfin

```
const axios = require('axios');
```

// Configuración de cliente

```
const jellyfinClient = axios.create({
  baseURL: 'http://localhost:8096/jellyfin',
  headers: {
    'X-Emby-Token': 'tu-api-key-aquí', // Token de API de Jellyfin
    'Content-Type': 'application/json'
  }
});
```

// Obtener Lista de usuarios

```
async function getJellyfinUsers() {
  try {
    const response = await jellyfinClient.get('/Users');
    return response.data;
  } catch (error) {
    console.error('Error al obtener usuarios de Jellyfin:', error);
    throw error;
  }
}
```

// Crear usuario (ejemplo simplificado)

```
async function createJellyfinUser(username, password) {
  try {
    const response = await jellyfinClient.post('/Users', {
      Name: username,
      Password: password
    });
    return response.data;
  } catch (error) {
    console.error('Error al crear usuario en Jellyfin:', error);
    throw error;
  }
}
```

// Exportar funciones

```
module.exports = {
  getJellyfinUsers,
  createJellyfinUser
};
```

14.3 Conexión a JFA-GO (SQLite)

// Ejemplo de conexión a base de datos de JFA-GO

```
const Database = require('better-sqlite3');
```

// Inicializar conexión

```
function connectToJfaGo(dbPath) {  
  try {  
    const db = new Database(dbPath, { readonly: false });  
    console.log('Conectado a base de datos JFA-GO');  
    return db;  
  } catch (error) {  
    console.error('Error al conectar con JFA-GO:', error);  
    throw error;  
  }  
}
```

// Obtener invitaciones activas

```
function getActiveInvitations(db) {  
  try {  
    const stmt = db.prepare('SELECT * FROM invites WHERE uses_remaining > 0');  
    return stmt.all();  
  } catch (error) {  
    console.error('Error al obtener invitaciones:', error);  
    throw error;  
  }  
}
```

// Crear invitación (ejemplo simplificado)

```
function createInvitation(db, expiry, usesRemaining, profile) {  
  try {  
    const stmt = db.prepare(`  
      INSERT INTO invites (code, created, expires, uses_remaining, profile_id)  
      VALUES (?, ?, ?, ?, ?)  
    `);  
  
    const code = generateRandomCode(); // Función para generar código único  
    const created = Math.floor(Date.now() / 1000); // Timestamp actual en segundos  
  
    const result = stmt.run(code, created, expiry, usesRemaining, profile);  
    return { code, id: result.lastInsertRowid };  
  } catch (error) {  
    console.error('Error al crear invitación:', error);  
    throw error;  
  }  
}
```

// Función auxiliar para generar código aleatorio

```
function generateRandomCode(length = 8) {
  const chars = 'ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789';
  let result = '';
  for (let i = 0; i < length; i++) {
    result += chars.charAt(Math.floor(Math.random() * chars.length));
  }
  return result;
}

module.exports = {
  connectToJfaGo,
  getActiveInvitations,
  createInvitation
};
```

15. Plantillas para Docker

15.1 Dockerfile para Backend

```
dockerfile

FROM node:18-alpine

WORKDIR /app

COPY package*.json ./

RUN npm install

COPY . .

EXPOSE 3000

CMD ["npm", "start"]
```

15.2 Dockerfile para Frontend

```
dockerfile
```

```
FROM node:18-alpine as build-stage
```

```
WORKDIR /app
```

```
COPY package*.json ./
```

```
RUN npm install
```

```
COPY . .
```

```
RUN npm run build
```

```
FROM nginx:stable-alpine as production-stage
```

```
COPY --from=build-stage /app/dist /usr/share/nginx/html
```

```
COPY nginx.conf /etc/nginx/conf.d/default.conf
```

```
EXPOSE 80
```

```
CMD ["nginx", "-g", "daemon off;"]
```

15.3 Configuración Nginx para Frontend (nginx.conf)

```
nginx
```

```
server {  
    listen 80;  
    server_name localhost;  
  
    root /usr/share/nginx/html;  
    index index.html;  
  
    location / {  
        try_files $uri $uri/ /index.html;  
    }  
  
    # Configuración para caché de archivos estáticos  
    location ~* \.(jpg|jpeg|png|gif|ico|css|js)$ {  
        expires 7d;  
        add_header Cache-Control "public, no-transform";  
    }  
}
```


16. Recursos Adicionales

16.1 Plantillas para Comunicaciones

Tipo	Propósito	Formato
Bienvenida	Nuevo cliente	Email, WhatsApp
Recordatorio de Pago	Factura pendiente	Email, SMS, WhatsApp
Problema Técnico	Notificación de incidencias	Email, SMS, Telegram
Invitación Jellyfin	Acceso a streaming	Email
Mantenimiento Programado	Aviso preventivo	Email, SMS, Telegram

16.2 Enlaces Útiles para WISP

Recurso	Descripción	Enlace
WISPA	Asociación de Proveedores de Internet Inalámbricos	wispa.org
Mikrotik Wiki	Documentación técnica	wiki.mikrotik.com
UBNT Community	Foro para equipos Ubiquiti	community.ui.com
WISP Talk	Foro especializado para WISP	wisptalk.com

17. Planificación de Recursos para el Proyecto (20 días)

17.1 Recursos Humanos

Rol	Dedicación	Habilidades Principales
Líder de Desarrollo	Tiempo completo	Node.js, arquitectura, Docker
Desarrollador Frontend	Tiempo completo	Vue.js, UI/UX, integraciones
Desarrollador Full-stack	Tiempo completo	JavaScript, bases de datos, API

17.2 Recursos Técnicos

Recurso	Cantidad	Propósito
Servidor de Desarrollo	1	Entorno de desarrollo
Servidor de Producción	1	Despliegue final
Router Mikrotik (pruebas)	1	Pruebas de integración
Dispositivo Ubiquiti (pruebas)	1	Pruebas de integración
Repositorio Git	1	Control de versiones

17.3 Herramientas de Gestión

Herramienta	Propósito	Enlace
GitHub/GitLab	Repositorio de código	github.com / gitlab.com
Trello	Gestión de tareas	trello.com
Discord/Slack	Comunicación del equipo	discord.com / slack.com
Google Drive	Compartir documentos	drive.google.com

Conclusión

Este documento proporciona una guía completa de las herramientas y recursos necesarios para implementar el Sistema Integral para ISP en 20 días. Utilizando estas tecnologías y recursos, el equipo podrá desarrollar una solución robusta, escalable y adaptada a las necesidades específicas de un proveedor de servicios de Internet, integrando eficientemente equipos Mikrotik, Ubiquiti, Jellyfin y múltiples servicios de comunicación.

La combinación de Node.js en el backend y Vue.js en el frontend ofrece un equilibrio óptimo entre facilidad de desarrollo, rendimiento y mantenibilidad, permitiendo el desarrollo rápido sin comprometer la calidad o escalabilidad futura del sistema.