📋 Guía Completa: Cómo Crear una Página Web Profesional como NICMAT S.R.L.

Fecha: Enero 2026
Versión: 1.0
Autor: Documentación técnica para NICMAT S.R.L.

ÍNDICE

Introducción

Esta guía explica cómo solicitar la creación de una página web profesional con todas las características de NICMAT S.R.L., incluyendo:

- ☑ Actualizaciones en tiempo real (como WhatsApp)
- ☑ Diseño responsivo para cualquier dispositivo
- ☑ Generación de PDFs profesionales
- ☑ Importación/exportación de Excel
- ☑ Sistema de usuarios y permisos
- ☑ Buenas prácticas de programación

Parte 1: Página con Realtime Básico

Qué decir al programador/IA:

> "Necesito una página web que funcione como WhatsApp o como la página de NICMAT. Es decir, quiero que cuando una persona hace un cambio en una tabla (en la laptop), inmediatamente ese cambio aparezca en OTRA PERSONA que está mirando la misma tabla (en la tablet) SIN que tenga que refrescar la página.
>
> Quiero que uses:
>
> - **React o Next.js** para la interfaz (para que sea rápido y moderno)
> - **Supabase** como base de datos con **Realtime habilitado** en todas las tablas
> - **Row Level Security desactivado** (porque mi aplicación usa autenticación custom)
> - **Vercel** para desplegar la página (así está siempre disponible)
> - **Anti-caché** en las APIs (agregar `force-dynamic` para que los datos siempre sean frescos)
> - Que las suscripciones a Realtime usen `useRef` para que siempre usen la función más actualizada
> - **Sin spinner de carga** en las actualizaciones de Realtime (que los datos aparezcan en silencio)
> - Logs visibles en la consola para debuggear si algo falla"

Parte 2: Aplicación Completa

Requisitos Generales

> "Necesito una aplicación web completa para gestionar un negocio, similar a NICMAT S.R.L. La página debe:

1. FUNCIONALIDAD GENERAL

- Tener un sistema de login con usuario y contraseña
- Que solo usuarios autenticados puedan entrar
- Que cada usuario tenga permisos diferentes (algunos ven todo, otros solo ven ciertas secciones)
- Que funcione en cualquier dispositivo: computadora, tablet, celular (diseño responsivo)
- Que sea rápida y no se quede congelada
- Que funcione en tiempo real como WhatsApp (cuando alguien hace un cambio, todos lo ven instantáneamente)

2. MÓDULOS/SECCIONES necesarios:

- **Inventario**: tabla para agregar productos, ver cantidades, editar, eliminar
- **Tiendas**: gestionar múltiples sucursales/tiendas
- **Cotizaciones**: crear presupuestos para clientes
- **Ventas**: registrar ventas de cada tienda
- **Gastos**: registrar gastos de cada tienda
- **Reportes/Estadísticas**: gráficos, totales, ranking de tiendas
- **Panel de Control (Dashboard)**: un resumen con los números principales del negocio
- **Usuarios**: crear usuarios, cambiar permisos

3. FUNCIONALIDADES ESPECÍFICAS

- Importar datos en Excel (cargar muchos productos de una vez)
- Exportar datos en Excel (descargar reportes)

- Generar PDFs profesionales (cotizaciones, remitos, facturas) que se vean bien impresos
- Buscar productos rápidamente
- Filtrar datos (por fecha, por tienda, por estado, etc.)
- Paginar datos (mostrar 10 o 50 registros por página)
- Ver el total, subtotal, ganancias en tiempo real
- Historial de cambios (quién cambió qué y cuándo)
- Modo oscuro/claro (tema de la página)"

Arquitectura y Tecnología

Frontend (lo que ve el usuario):

Tecnología	Propósito
Next.js 14+	Framework React moderno, rápido, con rutas dinámicas
TypeScript	Para evitar errores y que el código sea limpio
Tailwind CSS	Para que se vea bonito sin escribir mucho CSS
Shadcn/ui	Componentes profesionales listos para usar
React Context	Para guardar datos globales como quién es el usuario
Hooks	Para manejar estados y efectos
Realtime	WebSocket para actualizaciones instantáneas

Backend y Base de Datos:

Tecnología	Propósito
Supabase	Base de datos PostgreSQL + autenticación + Realtime
Next.js API Routes	Funciones del servidor para proteger datos sensibles
Middleware	Para verificar quién puede acceder a qué
Row Level Security desactivado	Porque usamos autenticación custom

Despliegue:

Servicio	Propósito
Vercel	Para desplegar la página y que esté disponible 24/7
GitHub	Para guardar el código y hacer cambios seguros
Environment Variables	Para secretos como contraseñas de base de datos

---

Tablas de Base de Datos

Estructura recomendada:

```
Tabla 'users':
- id, username, password, full_name, role (admin/usuario), is_active, empresa_id

Tabla 'inventory':
- id, marca, amperaje, cantidad, costo, precio_venta, created_at, updated_at

Tabla 'tiendas':
- id, nombre, tipo, ciudad, encargado, created_at, updated_at

Tabla 'tienda_inventario':
- id, tienda_id, marca, amperaje, cantidad, costo, precio_venta

Tabla 'tienda_ventas':
- id, tienda_id, fecha, total_venta, total_costo, ganancia

Tabla 'tienda_venta_items':
- id, venta_id, inventario_id, cantidad, precio_venta, costo

Tabla 'tienda_envios':
- id, tienda_id, estado, total_productos, total_unidades, created_at

Tabla 'tienda_envio_items':
- id, envio_id, inventario_id, cantidad, costo_original, precio_unitario

Tabla 'tienda_gastos':
- id, tienda_id, categoria, descripcion, monto, fecha

Tabla 'cotizaciones':
- id, numero, cliente_nombre, cliente_telefono, fecha, estado, total

Tabla 'empresa_config':
- id, nombre, rut, direction, ciudad, telefono_principal, telefono_secundario
```

IMPORTANTE: Todas las tablas deben tener **Realtime habilitado** en Supabase.

Diseño y Experiencia de Usuario

El diseño debe incluir:

- ☑ Menú a la izquierda con las opciones principales
- ☑ Header arriba con el nombre de la empresa y la foto del usuario
- ☑ Colores profesionales (azul, gris, blanco)
- ☑ Íconos para cada sección (inventario, tiendas, etc.)
- ☑ Tablas con datos limpias, con colores para estados (verde=activo, rojo=inactivo)
- ☑ Botones grandes y fáciles de clickear
- ☑ Modales (ventanas emergentes) para crear/editar datos
- ☑ Mensajes de confirmación cuando se borra algo
- ☑ Notificaciones cuando algo sale bien o mal
- ☑ Responsive: debe verse bien en celular, tablet, y computadora

Funcionalidades Avanzadas

Características requeridas:

Funcionalidad	Descripción
Búsqueda instantánea	Cuando escribo algo, filtra los datos en tiempo real
Validaciones	No dejar que ingrese datos inválidos (números negativos, campos vacíos)
Cálculos automáticos	Cuando ingreso cantidad y precio, que calcule el total automáticamente
Logs de auditoría	Guardar quién cambió qué y cuándo
Presencia online	Ver quién está usando la app en este momento
Anti-caché	Que los datos siempre estén frescos, no datos viejos
Actualizaciones silenciosas	Cuando llega un cambio por Realtime, que actualice sin mostrar spinner
Compresión de datos	Que la página sea rápida incluso con internet lento
Modo offline básico	Guardar últimos datos en el navegador por si se cae internet

Generación de Documentos

Para PDFs y Excel:

Excel:

- Usar librería `xlsx` para generar archivos Excel profesionales
- Los Excel deben tener múltiples hojas (una por sección)
- Validar que los datos no se desconfiguren al exportar

PDF:

- Usar `jspdf` o `html2pdf` para generar PDFs que se vean exactamente como en pantalla
- Los PDFs deben tener:
  - Logo de la empresa
  - Fecha
  - Datos del cliente
  - Tabla de productos
  - Totales
- Los PDFs deben ser descargables automáticamente

Seguridad

Implementar:

Medida	Descripción

Medida	Descripción
Contraseñas encriptadas	Nunca guardar contraseñas en texto plano
Autenticación custom	Cookies + tokens seguros
Middleware	Verificar que el usuario está autenticado antes de entrar a cualquier página
Validaciones en el servidor	No confiar solo en lo que envía el frontend
Variables de entorno	Contraseñas de base de datos nunca en el código
HTTPS	Comunicación encriptada
Rate limiting	Evitar ataques de fuerza bruta
Sanitización de datos	Evitar inyección SQL

Performance y Optimización

La página debe:

- Cargar en menos de 2 segundos
- Usar caché donde sea posible (pero con Realtime, no cachear datos sensibles)
- Optimizar imágenes
- Lazy loading (cargar contenido solo cuando se necesita)
- Code splitting (dividir el código en partes más pequeñas)
- Usar índices en la base de datos para búsquedas rápidas
- Paginación (no cargar todo de una vez)
- WebWorkers para cálculos pesados
- Compresión de JavaScript y CSS

Testing y Calidad

Implementar:

- **Tests unitarios** - Para funciones específicas
- **Tests de integración** - Para flujos completos
- **Linting** - ESLint para código limpio
- **Type checking** - TypeScript para atrapar errores
- **Error boundaries** - Para que si algo falla, no se quiebre todo
- **Logging de errores** - Guardar errores para debuggear después
- **Monitoring** - Alertas si algo está fallando en producción

---

Responsividad

Debe funcionar perfecto en:

Dispositivo	Características

Dispositivo	Características
Celulares	iPhone 12 en adelante, Android
Tablets	iPad, Samsung Tab
Laptops	Cualquier resolución
Televisores	Si alguien lo abre en una TV

Consideraciones adicionales:

- Touch-friendly (botones grandes para dedo)
- Teclado en móvil (inputs grandes y fáciles de escribir)

Despliegue y Mantenimiento

Configurar:

Servicio	Propósito
GitHub	Para controlar versiones del código
Vercel	Para desplegar automáticamente cuando hago push a GitHub
Environment variables	Para secretos
Backups automáticos	En Supabase
Monitoreo	Para saber si algo falla
Analytics	Para ver cuántas personas usan la app
CDN	Para distribuir contenido rápidamente

Documentación

El proyecto debe incluir:

- **README.md** - Explicar cómo instalar y usar
- **Comentarios en el código** - Explicando qué hace cada función
- **Guía de configuración** - Pasos para montar todo
- **Guía de usuario** - Cómo usar cada sección
- **API documentation** - Qué hace cada endpoint

Configuración de Realtime

Checklist para que Realtime funcione:

1. Tabla en publicación Realtime:

```
ALTER PUBLICATION supabase_realtime ADD TABLE nombre_tabla;
```

2. RLS desactivado o política permisiva:

```
-- Opción A: Desactivar RLS
ALTER TABLE nombre_tabla DISABLE ROW LEVEL SECURITY;

-- Opción B: Política permisiva
CREATE POLICY "Allow select" ON nombre_tabla FOR SELECT USING (true);
```

3. Cliente Supabase con config realtime:

```
createClient(url, key, {
  realtime: { params: { eventsPerSecond: 10 } }
});
```

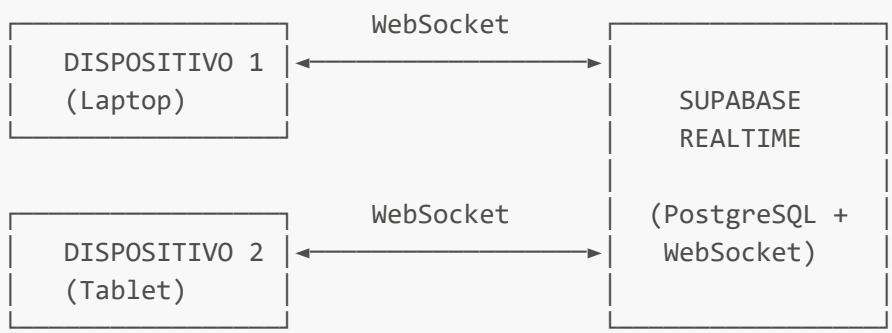
4. Suscripción a cambios:

```
supabase.channel('mi-canal')
  .on('postgres_changes', { event: '*', schema: 'public', table: 'mi_tabla' },
callback)
  .subscribe();
```

5. API sin caché (Next.js/Vercel):

```
export const dynamic = 'force-dynamic';
export const revalidate = 0;
```

Arquitectura de Realtime:



```
Cuando Laptop hace UPDATE → Supabase detecta → Envía evento a Tablet
```

Resumen Ejecutivo

Versión corta para solicitar:

> "Necesito una app web profesional como NICMAT S.R.L. que:
>
> 1. **Tenga login** y permisos de usuarios
> 2. **Funcione en tiempo real** (cambios instantáneos entre dispositivos)
> 3. **Sea responsiva** (móvil, tablet, PC)
> 4. **Tenga estos módulos**: Inventario, Tiendas, Cotizaciones, Ventas, Gastos, Estadísticas, Dashboard, Usuarios
> 5. **Importe/exporte Excel**
> 6. **Genere PDFs profesionales**
> 7. **Sea rápida y segura**
> 8. **Use Next.js + Supabase + Vercel**
> 9. **Esté deployada en la nube 24/7**
> 10. **Tenga buenas prácticas de programación**"

Preguntas Importantes

Preguntas que hacer al programador:

Pregunta	Por qué es importante
¿Cuánto cuesta?	Para presupuestar
¿Cuánto tiempo toma hacerlo?	Para planificar
¿Qué pasa si necesito cambios después?	Para saber el costo de mantenimiento
¿Me das el código o me das acceso a la app?	Para saber si eres dueño del código
¿Quién paga el hosting (Vercel, Supabase)?	Para calcular costos recurrentes
¿Cuántos usuarios simultáneos puede soportar?	Para saber si escala
¿Hay limite de datos que puedo guardar?	Para planificar crecimiento
¿Qué pasa si quiero agregar nuevas secciones después?	Para saber flexibilidad
¿Me enseñas a mantenerla?	Para independencia futura

Estado Actual de NICMAT S.R.L.

Páginas con Realtime activado:

Página	Tablas que escucha	Estado
Dashboard	inventory, cotizaciones, users, tienda_ventas	☑ Activo
Inventario	inventory	☑ Activo
Tiendas	tiendas, tienda_inventario, inventory, tienda_envios	☑ Activo
Cotizaciones	cotizaciones, empresa_config	☑ Activo
Estadísticas	tienda_ventas, tienda_inventario, tiendas	☑ Activo
Movimientos	tienda_ventas, tienda_gastos, cotizaciones, tienda_inventario	☑ Activo
Usuarios (Admin)	users, user_presence	☑ Activo

Contacto y Soporte

Para consultas sobre este documento o la aplicación NICMAT S.R.L., contactar al equipo de desarrollo.

Documento generado: Enero 2026
Última actualización: Enero 2026
Versión de la aplicación: 1.0.0