

🚀 Implementación del App Shell para NutriTrack Pro

Te ayudo a implementar un App Shell funcional basado en tu arquitectura. Aquí está el plan completo:

📁 Estructura Actualizada del Cliente

...

client/

- └─ public/
 - └─ index.html
 - └─ sw.js # Service Worker
 - └─ manifest.json # Web App Manifest
 - └─ icons/
 - └─ icon-192.png
 - └─ icon-512.png
 - └─ ...
- └─ src/
 - └─ App.vue # App Shell principal
 - └─ main.js
 - └─ components/
 - └─ AppHeader.vue # Encabezado
 - └─ AppFooter.vue # Pie de página
 - └─ AppMenu.vue # Menú principal
 - └─ LoadingSpinner.vue
 - └─ views/

```

| | | └─ Splash.vue    # Pantalla de inicio
| | | └─ Home.vue      # Vista principal con contenido dinámico
| | | └─ Diary.vue     # Otra vista de contenido
| | | └─ Settings.vue
| | └─ router/
| | └─ index.js        # Configuración de rutas
| | └─ stores/
| | └─ app.js          # Estado de la aplicación
| | └─ utils/
| └─ api.js            # Simulación de datos dinámicos
...

```

1. 🏗️ **Estructura del App Shell**

`src/App.vue` - App Shell Principal

```

```vue
<template>

 <div id="app" class="app-shell">

 <!-- Splash Screen -->

 <SplashScreen v-if="showSplash" @complete="hideSplash" />

 <!-- App Shell Principal -->

 <div v-else class="app-layout">

 <AppHeader @toggle-menu="toggleMenu" />

```

```
<div class="app-content">
```

```
 <AppMenu
```

```
 :is-open="menuOpen"
```

```
 @navigate="handleNavigation"
```

```
 @close="menuOpen = false"
```

```
 />
```

```
<main class="main-content" :class="{ 'menu-open': menuOpen }">
```

```
 <router-view v-slot="{ Component }">
```

```
 <transition name="fade" mode="out-in">
```

```
 <component :is="Component" />
```

```
 </transition>
```

```
 </router-view>
```

```
</main>
```

```
</div>
```

```
 <AppFooter />
```

```
</div>
```

```
</div>
```

```
</template>
```

```
<script setup>
```

```
import { ref, onMounted } from 'vue'
```

```
import { useRouter } from 'vue-router'
```

```
import SplashScreen from './views/Splash.vue'
```

```
import AppHeader from './components/AppHeader.vue'
import AppMenu from './components/AppMenu.vue'
import AppFooter from './components/AppFooter.vue'
```

```
const router = useRouter()
const showSplash = ref(true)
const menuOpen = ref(false)
```

```
const hideSplash = () => {
 showSplash.value = false
}
```

```
const toggleMenu = () => {
 menuOpen.value = !menuOpen.value
}
```

```
const handleNavigation = (route) => {
 router.push(route)
 menuOpen.value = false
}
```

```
// Registrar Service Worker al cargar la app
onMounted(() => {
 if ('serviceWorker' in navigator) {
 navigator.serviceWorker.register('/sw.js')
 .then(registration => console.log('SW registered: ', registration))
```

```
 .catch(error => console.log('SW registration failed: ', error))
 }
})
</script>
```

```
<style scoped>
```

```
.app-shell {
 height: 100vh;
 display: flex;
 flex-direction: column;
}
```

```
.app-layout {
 display: flex;
 flex-direction: column;
 height: 100%;
}
```

```
.app-content {
 display: flex;
 flex: 1;
 position: relative;
 overflow: hidden;
}
```

```
.main-content {
```

```
flex: 1;

transition: transform 0.3s ease;

overflow-y: auto;

padding: 1rem;
}
```

```
.main-content.menu-open {
 transform: translateX(250px);
}
```

```
.fade-enter-active,
.fade-leave-active {
 transition: opacity 0.3s ease;
}
```

```
.fade-enter-from,
.fade-leave-to {
 opacity: 0;
}
```

```
</style>
```

```
...
```

```
`src/components/AppHeader.vue`
```

```
```vue
```

```
<template>
```

```
  <header class="app-header">
```

```

<button class="menu-button" @click="$emit('toggle-menu')">

  <span class="hamburger">☰</span>

</button>

<h1 class="app-title">🍏 NutriTrack Pro</h1>

<div class="header-actions">

  <button class="icon-button" @click="toggleDarkMode">

    {{ darkMode ? '☀️' : '🌙' }}

  </button>

  <button class="icon-button" @click="refreshData">🔄 </button>

</div>

</header>

</template>

<script setup>

import { ref } from 'vue'

defineEmits(['toggle-menu'])

const darkMode = ref(false)

const toggleDarkMode = () => {

  darkMode.value = !darkMode.value

  document.body.classList.toggle('dark-mode', darkMode.value)

}

```

```
const refreshData = () => {  
  // Simular actualización de datos  
  window.dispatchEvent(new CustomEvent('refresh-data'))  
}  
</script>
```

```
<style scoped>  
.app-header {  
  display: flex;  
  align-items: center;  
  justify-content: space-between;  
  padding: 1rem;  
  background: linear-gradient(135deg, #22c55e, #16a34a);  
  color: white;  
  box-shadow: 0 2px 10px rgba(0,0,0,0.1);  
}
```

```
.menu-button {  
  background: none;  
  border: none;  
  color: white;  
  font-size: 1.5rem;  
  cursor: pointer;  
  padding: 0.5rem;  
}
```



```
.app-title {  
  font-size: 1.2rem;  
  font-weight: 600;  
  margin: 0;  
}
```

```
.header-actions {  
  display: flex;  
  gap: 0.5rem;  
}
```

```
.icon-button {  
  background: rgba(255,255,255,0.2);  
  border: none;  
  border-radius: 50%;  
  width: 2.5rem;  
  height: 2.5rem;  
  cursor: pointer;  
  font-size: 1rem;  
  display: flex;  
  align-items: center;  
  justify-content: center;  
  transition: background 0.3s;  
}
```

```
.icon-button:hover {
  background: rgba(255,255,255,0.3);
}
</style>
` ``
```

```
### **`src/components/AppMenu.vue`**
```

```
` `` vue
```

```
<template>
```

```
<aside v-if="isOpen" class="app-menu" @click="$emit('close')">
```

```
<nav class="menu-content" @click.stop>
```

```
<div class="menu-header">
```

```
<h2>Menú</h2>
```

```
<button class="close-button" @click="$emit('close')">x</button>
```

```
</div>
```

```
<ul class="menu-items">
```

```
<li v-for="item in menuItems" :key="item.route" class="menu-item">
```

```
<button
```

```
class="menu-link"
```

```
@click="$emit('navigate', item.route)"
```

```
:class="{ active: $route.path === item.route }"
```

```
>
```

```
<span class="menu-icon">{{ item.icon }}</span>
```

```
<span class="menu-text">{{ item.text }}</span>
```


```
</button>
```

```
</li>
```

```
</ul>
```

```
<div class="menu-footer">
```

```
<div class="user-info">
```

```
<div class="user-avatar">  </div>
```

```
<div class="user-details">
```

```
<strong>Alex García</strong>
```

```
<small>Objetivo: 2200 kcal</small>
```

```
</div>
```

```
</div>
```

```
</div>
```

```
</nav>
```

```
</aside>
```

```
</template>
```

```
<script setup>
```

```
import { computed } from 'vue'
```

```
import { useRoute } from 'vue-router'
```

```
defineProps({
```

```
  isOpen: Boolean
```

```
})
```

```
defineEmits(['navigate', 'close'])
```

```
const route = useRoute()
```

```
const menuItems = computed(() => [
```

```
  { icon: '🏠', text: 'Inicio', route: '/' },
```

```
  { icon: '📅', text: 'Mi Diario', route: '/diary' },
```

```
  { icon: '🍎', text: 'Alimentos', route: '/foods' },
```

```
  { icon: '🏋️', text: 'Ejercicios', route: '/workouts' },
```

```
  { icon: '⚙️', text: 'Configuración', route: '/settings' }
```

```
])
```

```
</script>
```

```
<style scoped>
```

```
.app-menu {
```

```
  position: fixed;
```

```
  top: 0;
```

```
  left: 0;
```

```
  width: 100%;
```

```
  height: 100%;
```

```
  background: rgba(0,0,0,0.5);
```

```
  z-index: 1000;
```

```
}
```

```
.menu-content {
```

```
  position: absolute;
```

```
  top: 0;
```

```
  left: 0;
```

```
width: 250px;
height: 100%;
background: white;
display: flex;
flex-direction: column;
box-shadow: 2px 0 10px rgba(0,0,0,0.1);
}
```

```
.menu-header {
display: flex;
justify-content: space-between;
align-items: center;
padding: 1rem;
background: #f8fafc;
border-bottom: 1px solid #e2e8f0;
}
```

```
.menu-header h2 {
margin: 0;
color: #1e293b;
}
```

```
.close-button {
background: none;
border: none;
font-size: 1.5rem;
```

```
    cursor: pointer;
    color: #64748b;
}
```

```
.menu-items {
    flex: 1;
    list-style: none;
    padding: 0;
    margin: 0;
}
```

```
.menu-item {
    border-bottom: 1px solid #f1f5f9;
}
```

```
.menu-link {
    display: flex;
    align-items: center;
    width: 100%;
    padding: 1rem;
    background: none;
    border: none;
    text-align: left;
    cursor: pointer;
    transition: background 0.3s;
}
```

```
.menu-link:hover {  
  background: #f1f5f9;  
}
```

```
.menu-link.active {  
  background: #dcfce7;  
  color: #16a34a;  
}
```

```
.menu-icon {  
  font-size: 1.2rem;  
  margin-right: 0.75rem;  
}
```

```
.menu-text {  
  font-weight: 500;  
}
```

```
.menu-footer {  
  padding: 1rem;  
  background: #f8fafa;  
  border-top: 1px solid #e2e8f0;  
}
```

```
.user-info {
```

```
display: flex;
align-items: center;
gap: 0.75rem;
}
```

```
.user-avatar {
width: 2.5rem;
height: 2.5rem;
background: #22c55e;
border-radius: 50%;
display: flex;
align-items: center;
justify-content: center;
color: white;
}
```

```
.user-details {
flex: 1;
}
```

```
.user-details strong {
display: block;
font-size: 0.9rem;
}
```

```
.user-details small {
```



```
    color: #64748b;

    font-size: 0.75rem;
  }
</style>
` ``
```

```
### **`src/components/AppFooter.vue`**
```

```
` `` vue
```

```
<template>
```

```
  <footer class="app-footer">
```

```
    <div class="footer-content">
```

```
      <p>&copy; 2024 NutriTrack Pro - Tu compañero de nutrición</p>
```

```
      <div class="footer-status">
```

```
        <span class="status-indicator" :class="connectionStatus">
```

```
          {{ connectionText }}
```

```
        </span>
```

```
      </div>
```

```
    </div>
```

```
  </footer>
```

```
</template>
```

```
<script setup>
```

```
import { ref, onMounted, onUnmounted } from 'vue'
```

```
const connectionStatus = ref('online')
```

```
const connectionText = ref('En línea')
```

```
const updateConnectionStatus = () => {  
  if (navigator.onLine) {  
    connectionStatus.value = 'online'  
    connectionText.value = 'En línea'  
  } else {  
    connectionStatus.value = 'offline'  
    connectionText.value = 'Sin conexión'  
  }  
}
```

```
onMounted(() => {  
  updateConnectionStatus()  
  window.addEventListener('online', updateConnectionStatus)  
  window.addEventListener('offline', updateConnectionStatus)  
})
```

```
onUnmounted(() => {  
  window.removeEventListener('online', updateConnectionStatus)  
  window.removeEventListener('offline', updateConnectionStatus)  
})
```

```
</script>
```

```
<style scoped>
```

```
.app-footer {  
  background: #1e293b;
```

```
color: white;

padding: 1rem;

border-top: 1px solid #334155;
}
```

```
.footer-content {

display: flex;

justify-content: space-between;

align-items: center;

max-width: 1200px;

margin: 0 auto;
}
```

```
.footer-content p {

margin: 0;

font-size: 0.875rem;
}
```

```
.status-indicator {

padding: 0.25rem 0.75rem;

border-radius: 1rem;

font-size: 0.75rem;

font-weight: 500;
}
```

```
.status-indicator.online {
```

```
background: #22c55e;
}
```

```
.status-indicator.offline {
  background: #ef4444;
}
```

```
</style>
```

```
```
```

```

```

## 2. 🛠️ \*\*Service Worker para App Shell\*\*

```
` public/sw.js `
```

```
```javascript
```

```
// Cache names
```

```
const CACHE_SHELL = 'nutritrack-shell-v1'
```

```
const CACHE_CONTENT = 'nutritrack-content-v1'
```

```
// App Shell assets to cache immediately
```

```
const SHELL_ASSETS = [
```

```
  '/',
```

```
  '/manifest.json',
```

```
  '/icons/icon-192.png',
```

```
  '/icons/icon-512.png'
```

```
]
```

```

// Content assets to cache on demand

const CONTENT_ASSETS = [

  // Estas se cachearán dinámicamente

]


// Install event - cache App Shell

self.addEventListener('install', (event) => {

  console.log('Service Worker: Installing...')


  event.waitUntil(

    caches.open(CACHE_SHELL)

    .then(cache => {

      console.log('Service Worker: Caching App Shell')

      return cache.addAll(SHELL_ASSETS)

    })

    .then(() => {

      console.log('Service Worker: Install completed')

      return self.skipWaiting()

    })

  )

})


// Activate event - cleanup old caches

self.addEventListener('activate', (event) => {

  console.log('Service Worker: Activating...')

```

```

event.waitUntil(
  caches.keys().then(cacheNames => {
    return Promise.all(
      cacheNames.map(cacheName => {
        if (cacheName !== CACHE_SHELL && cacheName !== CACHE_CONTENT) {
          console.log('Service Worker: Deleting old cache', cacheName)
          return caches.delete(cacheName)
        }
      })
    )
  }).then(() => {
    console.log('Service Worker: Activate completed')
    return self.clients.claim()
  })
)
})

```

// Fetch event - Serve from cache, fallback to network

```
self.addEventListener('fetch', (event) => {
```

```
  // Skip non-GET requests
```

```
  if (event.request.method !== 'GET') return
```

```
  // Skip Chrome extensions
```

```
  if (event.request.url.startsWith('chrome-extension://')) return
```

```
event.respondWith(
  caches.match(event.request)
    .then(response => {
      // Return cached version or fetch from network
      return response || fetch(event.request)
        .then(fetchResponse => {
          // Check if we received a valid response
          if (!fetchResponse || fetchResponse.status !== 200 || fetchResponse.type !==
'basic') {
            return fetchResponse
          }

          // Clone the response
          const responseToCache = fetchResponse.clone()

          // Cache the fetched resource
          caches.open(CACHE_CONTENT)
            .then(cache => {
              cache.put(event.request, responseToCache)
            })

          return fetchResponse
        })
    })
    .catch(error => {
      console.log('Fetch failed; returning offline page:', error)
    })
  )
)
```

```

    // If it's a navigation request and we're offline, serve the shell
    if (event.request.mode === 'navigate') {
        return caches.match('/')
    }

    // You could return a custom offline page here
    return new Response('Network error happened', {
        status: 408,
        headers: { 'Content-Type': 'text/plain' }
    })
})
})

// Background sync for offline data
self.addEventListener('sync', (event) => {
    if (event.tag === 'background-sync') {
        console.log('Background sync triggered')
        event.waitUntil(doBackgroundSync())
    }
})

async function doBackgroundSync() {
    // Aquí iría la lógica para sincronizar datos pendientes
    console.log('Syncing background data...')
}

```



```
}  
` ``
```

```
---
```

3. 🧩 **Web App Manifest**

```
### **` public/manifest.json `**
```

```
` `` `json
```

```
{
```

```
  "name": "NutriTrack Pro",
```

```
  "short_name": "NutriTrack",
```

```
  "description": "Tu compañero de nutrición y ejercicio - PWA",
```

```
  "start_url": "/",
```

```
  "display": "standalone",
```

```
  "background_color": "#22c55e",
```

```
  "theme_color": "#22c55e",
```

```
  "orientation": "portrait-primary",
```

```
  "scope": "/",
```

```
  "lang": "es-ES",
```

```
  "categories": ["health", "fitness", "lifestyle"],
```

```
  "icons": [  
    {
```

```
      {
```

```
        "src": "/icons/icon-72.png",
```

```
        "sizes": "72x72",
```

```
"type": "image/png",
"purpose": "any maskable"
},
{
  "src": "/icons/icon-96.png",
  "sizes": "96x96",
  "type": "image/png",
  "purpose": "any maskable"
},
{
  "src": "/icons/icon-128.png",
  "sizes": "128x128",
  "type": "image/png",
  "purpose": "any maskable"
},
{
  "src": "/icons/icon-144.png",
  "sizes": "144x144",
  "type": "image/png",
  "purpose": "any maskable"
},
{
  "src": "/icons/icon-152.png",
  "sizes": "152x152",
  "type": "image/png",
  "purpose": "any maskable"
```

```
},  
{  
  "src": "/icons/icon-192.png",  
  "sizes": "192x192",  
  "type": "image/png",  
  "purpose": "any maskable"  
},  
{  
  "src": "/icons/icon-384.png",  
  "sizes": "384x384",  
  "type": "image/png",  
  "purpose": "any maskable"  
},  
{  
  "src": "/icons/icon-512.png",  
  "sizes": "512x512",  
  "type": "image/png",  
  "purpose": "any maskable"  
}  
],
```

```
"screenshots": [  
  {  
    "src": "/screenshots/mobile.png",  
    "sizes": "375x667",  
    "type": "image/png",
```

```

    "form_factor": "narrow"
  },
  {
    "src": "/screenshots/desktop.png",
    "sizes": "1280x720",
    "type": "image/png",
    "form_factor": "wide"
  }
]
}
...

```

4. 🇮🇹 ****Vista con Contenido Dinámico****

```
### **`src/views/Home.vue`**
```

```
` ``vue
```

```
<template>
```

```
<div class="home-view">
```

```
<div class="welcome-section">
```

```
<h1>¡Bienvenido, Alex! 🙌 </h1>
```

```
<p>Hoy es {{ currentDate }}</p>
```

```
</div>
```

```
<!-- Resumen Nutricional -->
```

```
<section class="nutrition-summary">
```

```
<h2> 🍽️ Resumen del Día</h2>
```

```
<div class="stats-grid">
```

```
<div class="stat-card">
```

```
<div class="stat-value">{{ dailyStats.calories }}</div>
```

```
<div class="stat-label">Calorías</div>
```

```
<div class="stat-progress">
```

```
<div class="progress-bar">
```

```
<div class="progress-fill" :style="{ width: calorieProgress + '%' }"></div>
```

```
</div>
```

```
<span>{{ Math.round(calorieProgress) }}%</span>
```

```
</div>
```

```
</div>
```

```
<div class="stat-card">
```

```
<div class="stat-value">{{ dailyStats.protein }}g</div>
```

```
<div class="stat-label">Proteína</div>
```

```
<div class="stat-progress">
```

```
<div class="progress-bar">
```

```
<div class="progress-fill protein" :style="{ width: proteinProgress + '%' }"></div>
```

```
</div>
```

```
<span>{{ Math.round(proteinProgress) }}%</span>
```

```
</div>
```

```
</div>
```

```
</div>
```

```
</section>
```

```
<!-- Comidas Recientes -->

<section class="recent-meals">

  <div class="section-header">

    <h2>🍲 Comidas Recientes</h2>

    <button class="add-button" @click="addMockMeal">+ Agregar</button>

  </div>

  <div v-if="recentMeals.length === 0" class="empty-state">

    <p>No hay comidas registradas hoy</p>

    <button @click="addMockMeal" class="btn-primary">Agregar primera
comida</button>

  </div>

  <div v-else class="meals-list">

    <div v-for="meal in recentMeals" :key="meal.id" class="meal-card">

      <div class="meal-header">

        <span class="meal-type">{{ meal.type }}</span>

        <span class="meal-time">{{ meal.time }}</span>

      </div>

      <div class="meal-content">

        <h3>{{ meal.name }}</h3>

        <div class="meal-nutrition">

          <span>{{ meal.calories }} kcal</span>

          <span>{{ meal.protein }}g proteína</span>

        </div>

      </div>

    </div>

  </div>

</section>
```

```

    </div>

    <button class="delete-btn" @click="deleteMeal(meal.id)"> 🗑️ </button>

  </div>

</div>

</section>

<!-- Recordatorios -->

<section class="reminders">

  <h2> 🕒 Recordatorios</h2>

  <div class="reminders-list">

    <div v-for="reminder in reminders" :key="reminder.id" class="reminder-card">

      <span class="reminder-icon">{{ reminder.icon }}</span>

      <span class="reminder-text">{{ reminder.text }}</span>

      <button class="dismiss-btn"
        @click="dismissReminder(reminder.id)"> ✓ </button>

    </div>

  </div>

</section>

</div>

</template>

<script setup>

import { ref, computed, onMounted, onUnmounted } from 'vue'

// Estado reactivo

const recentMeals = ref([])

```

```
const reminders = ref([
  { id: 1, icon: '💧', text: 'Toma un vaso de agua', time: '10:00' },
  { id: 2, icon: '🏃', text: 'Camina 10 minutos', time: '12:00' },
  { id: 3, icon: '🍏', text: 'Registra tu almuerzo', time: '14:00' }
])
```

```
// Estadísticas del día
```

```
const dailyStats = computed(() => {
  const total = recentMeals.value.reduce((acc, meal) => ({
    calories: acc.calories + meal.calories,
    protein: acc.protein + meal.protein
  }), { calories: 0, protein: 0 })

  return total
})
```

```
// Progreso calculado
```

```
const calorieProgress = computed(() => (dailyStats.value.calories / 2200) * 100)
const proteinProgress = computed(() => (dailyStats.value.protein / 150) * 100)
```

```
// Fecha actual formateada
```

```
const currentDate = computed(() => {
  return new Date().toLocaleDateString('es-ES', {
    weekday: 'long',
    year: 'numeric',
    month: 'long',
  })
})
```



```
      day: 'numeric'
    })
  })

// Simular datos iniciales
const loadMockData = () => {
  const mockMeals = [
    {
      id: 1,
      type: 'Desayuno',
      time: '08:30',
      name: 'Yogur con granola y frutos rojos',
      calories: 320,
      protein: 15
    },
    {
      id: 2,
      type: 'Snack',
      time: '11:00',
      name: 'Manzana y almendras',
      calories: 180,
      protein: 6
    }
  ]
  recentMeals.value = mockMeals
}
```

// Agregar comida de ejemplo

```
const addMockMeal = () => {  
  const newMeal = {  
    id: Date.now(),  
    type: 'Comida',  
    time: new Date().toLocaleTimeString('es-ES', { hour: '2-digit', minute: '2-digit' }),  
    name: 'Pechuga de pollo con ensalada',  
    calories: 450,  
    protein: 35  
  }  
  recentMeals.value.unshift(newMeal)  
}
```

// Eliminar comida

```
const deleteMeal = (mealId) => {  
  recentMeals.value = recentMeals.value.filter(meal => meal.id !== mealId)  
}
```

// Descartar recordatorio

```
const dismissReminder = (reminderId) => {  
  reminders.value = reminders.value.filter(reminder => reminder.id !== reminderId)  
}
```

// Escuchar evento de actualización

```
const handleRefresh = () => {
```

```
    console.log('Actualizando datos...')

    loadMockData()
  }

  onMounted(() => {
    loadMockData()

    window.addEventListener('refresh-data', handleRefresh)
  })

  onUnmounted(() => {
    window.removeEventListener('refresh-data', handleRefresh)
  })
</script>
```

```
<style scoped>

.home-view {
  max-width: 800px;
  margin: 0 auto;
  padding-bottom: 2rem;
}
```

```
.welcome-section {
  text-align: center;
  margin-bottom: 2rem;
  padding: 1rem;
  background: linear-gradient(135deg, #22c55e, #16a34a);
```

```
    color: white;

    border-radius: 1rem;
}
```

```
.welcome-section h1 {

    margin: 0 0 0.5rem 0;

    font-size: 1.5rem;
}
```

```
.welcome-section p {

    margin: 0;

    opacity: 0.9;
}
```

```
section {

    margin-bottom: 2rem;
}
```

```
section h2 {

    margin-bottom: 1rem;

    color: #1e293b;

    font-size: 1.25rem;
}
```

```
.stats-grid {

    display: grid;
}
```

```
    grid-template-columns: 1fr 1fr;
    gap: 1rem;
}
```

```
.stat-card {
    background: white;
    padding: 1.5rem;
    border-radius: 1rem;
    box-shadow: 0 2px 10px rgba(0,0,0,0.1);
    text-align: center;
}
```

```
.stat-value {
    font-size: 2rem;
    font-weight: bold;
    color: #22c55e;
    margin-bottom: 0.5rem;
}
```

```
.stat-label {
    color: #64748b;
    margin-bottom: 1rem;
}
```

```
.stat-progress {
    display: flex;
```

```
align-items: center;
gap: 0.5rem;
}
```

```
.progress-bar {
  flex: 1;
  height: 8px;
  background: #e2e8f0;
  border-radius: 4px;
  overflow: hidden;
}
```

```
.progress-fill {
  height: 100%;
  background: #22c55e;
  transition: width 0.3s ease;
}
```

```
.progress-fill.protein {
  background: #3b82f6;
}
```

```
.section-header {
  display: flex;
  justify-content: space-between;
  align-items: center;
```

```
margin-bottom: 1rem;  
}
```

```
.add-button {  
  background: #22c55e;  
  color: white;  
  border: none;  
  padding: 0.5rem 1rem;  
  border-radius: 0.5rem;  
  cursor: pointer;  
  font-weight: 500;  
}
```

```
.empty-state {  
  text-align: center;  
  padding: 3rem 1rem;  
  background: white;  
  border-radius: 1rem;  
  box-shadow: 0 2px 10px rgba(0,0,0,0.1);  
}
```

```
.btn-primary {  
  background: #22c55e;  
  color: white;  
  border: none;  
  padding: 0.75rem 1.5rem;
```

```
border-radius: 0.5rem;  
cursor: pointer;  
font-weight: 500;  
margin-top: 1rem;  
}
```

```
.meals-list {  
  display: flex;  
  flex-direction: column;  
  gap: 1rem;  
}
```

```
.meal-card {  
  background: white;  
  padding: 1rem;  
  border-radius: 1rem;  
  box-shadow: 0 2px 10px rgba(0,0,0,0.1);  
  display: flex;  
  justify-content: between;  
  align-items: center;  
}
```

```
.meal-header {  
  display: flex;  
  flex-direction: column;  
  min-width: 80px;
```



```
}
```

```
.meal-type {  
  font-weight: 600;  
  color: #1e293b;  
}
```

```
.meal-time {  
  font-size: 0.875rem;  
  color: #64748b;  
}
```

```
.meal-content {  
  flex: 1;  
  margin: 0 1rem;  
}
```

```
.meal-content h3 {  
  margin: 0 0 0.5rem 0;  
  font-size: 1rem;  
}
```

```
.meal-nutrition {  
  display: flex;  
  gap: 1rem;  
  font-size: 0.875rem;
```

```
color: #64748b;  
}
```

```
.delete-btn {  
  background: none;  
  border: none;  
  cursor: pointer;  
  font-size: 1.2rem;  
  padding: 0.5rem;  
}
```

```
.reminders-list {  
  display: flex;  
  flex-direction: column;  
  gap: 0.5rem;  
}
```

```
.reminder-card {  
  background: white;  
  padding: 1rem;  
  border-radius: 0.75rem;  
  box-shadow: 0 2px 10px rgba(0,0,0,0.1);  
  display: flex;  
  align-items: center;  
  gap: 1rem;  
}
```

```
.reminder-icon {  
  font-size: 1.2rem;  
}
```

```
.reminder-text {  
  flex: 1;  
}
```

```
.dismiss-btn {  
  background: #22c55e;  
  color: white;  
  border: none;  
  width: 2rem;  
  height: 2rem;  
  border-radius: 50%;  
  cursor: pointer;  
  display: flex;  
  align-items: center;  
  justify-content: center;  
}
```

```
/* Responsive */
```

```
@media (max-width: 768px) {  
  .stats-grid {  
    grid-template-columns: 1fr;
```

```
}
```

```
.meal-card {  
  flex-direction: column;  
  align-items: stretch;  
  gap: 1rem;  
}
```

```
.meal-header {  
  flex-direction: row;  
  justify-content: space-between;  
}
```

```
}
```

```
</style>
```

```
...
```

```
---
```

5. 📖 ****Configuración de Rutas****

```
### **`src/router/index.js`**
```

```
` `` `javascript
```

```
import { createRouter, createWebHistory } from 'vue-router'
```

```
import Home from '../views/Home.vue'
```

```
import Diary from '../views/Diary.vue'
```

```
import Settings from '../views/Settings.vue'
```

```
const routes = [  
  {  
    path: '/',  
    name: 'Home',  
    component: Home  
  },  
  {  
    path: '/diary',  
    name: 'Diary',  
    component: Diary  
  },  
  {  
    path: '/settings',  
    name: 'Settings',  
    component: Settings  
  }  
]
```

```
const router = createRouter({  
  history: createWebHistory(),  
  routes  
})
```

```
export default router  
` ``
```

6. **Actualización del README**

Agrega esta sección a tu `README.md` :

` `` markdown

App Shell - Funcionalidad Offline

Estructura del App Shell

- **AppHeader**: Encabezado con menú hamburguesa y acciones
- **AppMenu**: Menú lateral deslizante con navegación
- **AppFooter**: Pie de página con estado de conexión
- **Vistas Dinámicas**: Home, Diary, Settings con contenido actualizable

Service Worker Configuration

El Service Worker (` public/sw.js `) implementa:

- **Cache First** para el App Shell
- **Network First** para contenido dinámico
- **Fallback offline** para navegación
- **Background sync** preparado para datos pendientes

Cómo Probar la Funcionalidad Offline

1. **Instalar la PWA**:

```
` `` bash
```

```
npm run dev
```

```
` ``
```

Abre <http://localhost:5173> y instala la app desde el navegador

2. ****Probar offline****:

- Abre DevTools → Application → Service Worker
- Haz clic en "Offline"
- Recarga la página - el App Shell debe cargar desde cache

3. ****Ver contenido cacheado****:

- DevTools → Application → Cache Storage
- Verifica los caches `nutritrack-shell-v1` y `nutritrack-content-v1`

Características PWA Implementadas

- ✓ App Shell arquitectura
- ✓ Service Worker con estrategias de cache
- ✓ Web App Manifest con metadatos
- ✓ Contenido dinámico simulando API
- ✓ Diseño responsive y accesible
- ✓ Estados de carga y errores
- ✓ Indicador de conexión
- ✓ Instalable como app nativa
- ✓ Funcionamiento offline completo

```
` ``
```

🎯 ****Próximos Pasos de Implementación****

1. ****Ejecuta el proyecto:****

```
` `` bash
```

```
npm run dev
```

```
` ``
```

2. ****Prueba la funcionalidad offline**** en DevTools

3. ****Verifica el manifest**** en DevTools → Application → Manifest

4. ****Instala la PWA**** desde el menú del navegador