

Documentação Técnica

Missale App - TEMPORA

Análise Completa da Estrutura e Funcionamento

1. Visão Geral do Aplicativo

O Missale App (também chamado TEMPORA) é um aplicativo híbrido desenvolvido com **Apache Cordova** para exibir o Missal Romano em múltiplos idiomas. O app funciona como uma WebView que renderiza conteúdo HTML/CSS/JavaScript, permitindo seu uso tanto em dispositivos móveis (Android/iOS) quanto em navegadores web.

1.1. Tecnologias Utilizadas

- **Framework:** Apache Cordova (aplicativo híbrido)
- **Frontend:** HTML5, CSS3, JavaScript (ES6)
- **Bibliotecas JavaScript:** jQuery, jQueryUI, iScroll-lite
- **Calendário:** JSCalendar (biblioteca customizada)
- **Armazenamento:** LocalStorage para preferências do usuário
- **Plugins Cordova:**
 - cordova-plugin-device: Informações do dispositivo
 - cordova-plugin-file: Acesso ao sistema de arquivos
 - cordova-plugin-globalization: Internacionalização
 - cordova-plugin-insomnia: Manter tela ligada
 - cordova-plugin-wkwebview-engine: Motor WebKit para iOS

2. Estrutura de Diretórios

A pasta /www contém todo o código-fonte da aplicação web:

```
www/
├── index.html          # Ponto de entrada (redireciona)
├── cordova.js          # Framework Cordova
├── cordova_plugins.js  # Registro dos plugins
├── jquery.js            # Biblioteca jQuery
├── jqueryui.js          # Biblioteca jQueryUI
├── jqueryui.css         # Estilos do jQueryUI
├── iscroll-lite.js      # Biblioteca de scroll suave
├── autocolumn.js        # Layout em colunas
├── ayudaweb.html        # Ajuda para versão web
└── css/
    └── index.css         # Estilos iniciais
└── js/
    └── index.js          # JavaScript inicial do Cordova
└── plugins/             # Plugins nativos do Cordova
    ├── cordova-plugin-device/
    ├── cordova-plugin-file/
    ├── cordova-plugin-globalization/
    └── cordova-plugin-insomnia/
└── misal_v2/             # CONTEÚDO PRINCIPAL DO APP
```

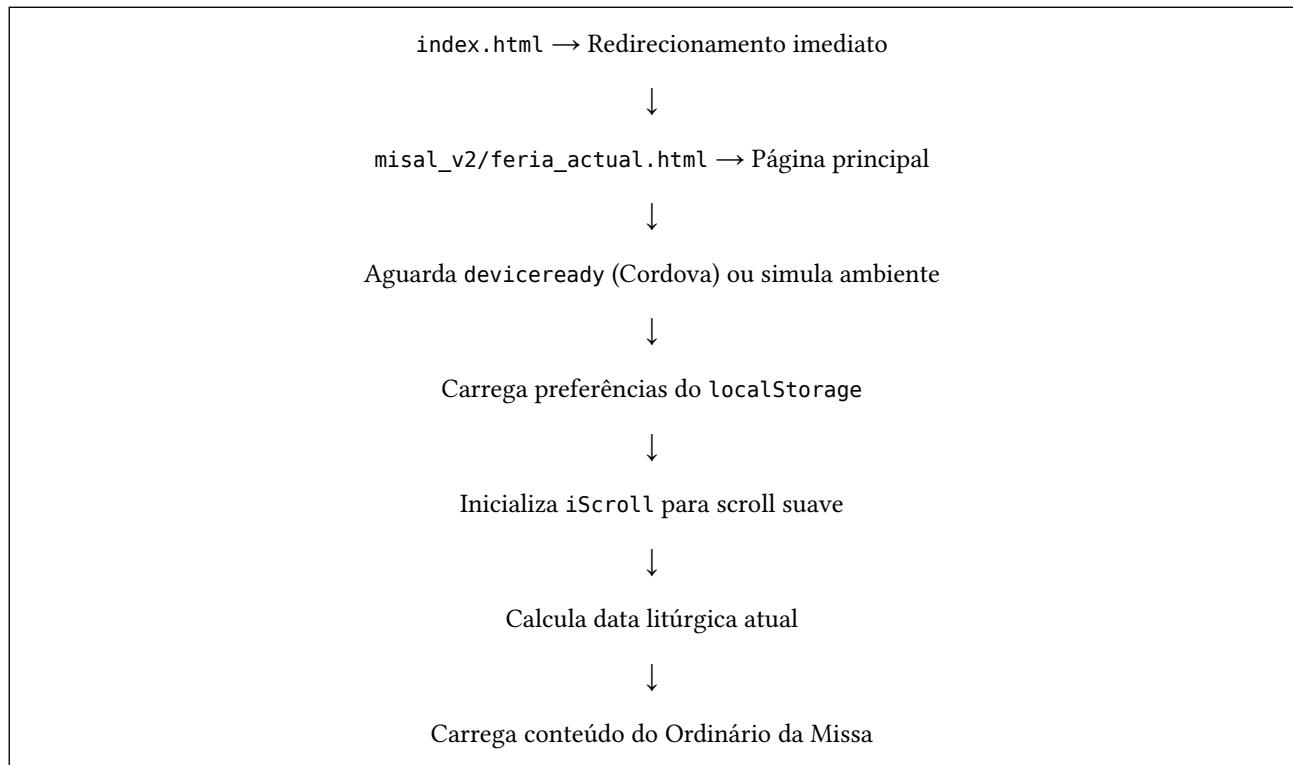
2.1. Estrutura da Pasta misal_v2/

Esta é a pasta mais importante, contendo todo o conteúdo litúrgico:

```
misal_v2/
├── feria_actual.html      # Página principal (calendário)
├── misal_todo.html        # Visualização completa do missal
├── preferencias.html      # Configurações do usuário
├── devocionario.html      # Devocionário
├── ayuda.html              # Ajuda
├── politica_misal.html    # Política de privacidade
├── misal.css                # Estilos principais
├── devoc.css                # Estilos do devocionário
├── mis_funciones_misal.js   # Funções principais (3691 linhas)
├── mis_funciones_devoc.js   # Funções do devocionário
├── jscalendar/             # Biblioteca de calendário
├── igmr/                   # Instrução Geral do Missal Romano
├── sacerdotale/             # Orações do sacerdote
├── m_estructura/           # Estrutura base (template)
├── m_cast/                  # Conteúdo em Castelhano
├── m_latin/                 # Conteúdo em Latim
├── m_port/                  # Conteúdo em Português
├── m_engl/                  # Conteúdo em Inglês
├── m_fran/                  # Conteúdo em Francês
├── m_ital/                  # Conteúdo em Italiano
└── m_germ/                  # Conteúdo em Alemão
```

3. Fluxo de Inicialização

O aplicativo segue este fluxo de inicialização:



Listagem 1: Fluxo de inicialização do aplicativo

3.1. Arquivo index.html

Este é o ponto de entrada do Cordova, mas apenas redireciona:

```
<!DOCTYPE html>
<html>
<head>
    <meta charset="utf-8" />
    <title>Iniciando Missal...</title>
    <script>
        window.location.href = "misal_v2/feria_actual.html";
    </script>
</head>
<body></body>
</html>
```

4. Arquivo Principal: `feria_actual.html`

Este arquivo (6339 linhas) é o coração do aplicativo, contendo:

4.1. Estrutura HTML

```
<body>
  <!-- Cabeçalho fixo com título -->
  <div id="header">
    <div id="cabecera">
      <h1>TEMPORA</h1>
      <nav class="navbar">...</nav>
    </div>
  </div>

  <!-- Container principal com scroll -->
  <div id="contenedor">
    <div id="scroller">
      <!-- Conteúdo dinâmico carregado aqui -->
      <div id="tab_o">...</div>  <!-- Ordinário -->
      <div id="tab_t">...</div>  <!-- Tempo litúrgico -->
      <!-- Outras abas -->
    </div>
  </div>

  <!-- Rodapé fixo com navegação -->
  <div id="piedepantalla">
    <div class="navbar">...</div>
  </div>
</body>
```

4.2. Simulação de Ambiente Móvel

Para funcionar em navegadores, o arquivo simula o ambiente Cordova:

```
var esDispositivoMovil = false;
window.cordova = undefined;

// Simula plugins
window.plugins = { insomnia: { keepAwake: function(){}} };
navigator.app = { exitApp: function(){ console.log("Sair"); } };

// Simula informações do dispositivo
if (!window.device) {
  window.device = {
    platform: 'Browser',
    version: '1.0',
    manufacturer: 'PC',
    model: 'Chrome'
  };
}
```

5. Sistema de Calendário Litúrgico

O app implementa um complexo sistema de cálculo do calendário litúrgico católico:

5.1. Funções de Cálculo de Datas

```
// Calcula a data da Páscoa (algoritmo de Gauss)
function domingo_pascua(anio) { ... }

// Calcula datas relacionadas à Páscoa
var fpascua = domingo_pascua(mifecha0.anio);
var fbautismo = bautismo(mifecha0.anio);
var fdomladv = prim_adviento(mifecha0.anio);
var framos = suma_dias_a_fecha(fpascua, -7);
var fceniza = suma_dias_a_fecha(fpascua, -46);
var fascension = suma_dias_a_fecha(fpascua, 40);
var fpentecostes = suma_dias_a_fecha(fpascua, 49);
var ftrinidad = suma_dias_a_fecha(fpascua, 56);
var fcorpus = suma_dias_a_fecha(fpascua, 63);
```

5.2. Ciclos Litúrgicos

O sistema calcula automaticamente o ciclo litúrgico (A, B ou C):

```
var ciclos = ['A', 'B', 'C'];
var tiposanio = ['I', 'II']; // Anos pares/ímpares

if (DiferenciaFechas(CadenaFecha, fdomladv) >= 0) {
    aniociclo = eval(mifecha0.anio) + 1;
} else {
    aniociclo = mifecha0.anio;
}

micinglo = ciclos[(aniociclo - 1969) % 3];
mitipoanio = tiposanio[aniociclo % 2];
```

5.3. Tempos Litúrgicos

O calendário identifica os seguintes períodos:

- **Advento:** 4 semanas antes do Natal
- **Natal:** 25 de dezembro até Batismo do Senhor
- **Tempo Ordinário I:** Após Batismo até Quaresma
- **Quaresma:** Quarta-feira de Cinzas até Páscoa
- **Tríduo Pascal:** Quinta, Sexta e Sábado Santos
- **Páscoa:** Domingo de Páscoa até Pentecostes
- **Tempo Ordinário II:** Após Pentecostes até Advento

6. Sistema de Idiomas

O aplicativo suporta 8 idiomas, cada um em sua própria pasta:

Código	Idioma	Pasta
cast	Castelhano (Espanhol)	m_cast/
latin	Latim	m_latin/
port	Português	m_port/
engl	Inglês	m_engl/
fran	Francês	m_fran/
ital	Italiano	m_ital/
germ	Alemão	m_germ/
estructura	Template/Estrutura	m_estructura/

6.1. Estrutura de Cada Pasta de Idioma

Cada pasta de idioma segue a mesma estrutura:

```
m_cast/ (exemplo para Castelhano)
├── ordinario/
│   ├── m_cast_ordinario.html      # Ordinário da Missa
│   ├── m_cast_bendiciones.html    # Bênçãos
│   ├── m_cast_oracion_fieles.html # Oração dos Fiéis
│   └── m_cast_oraciones_pueblo.html # Respostas do povo
└── tiempos/
    ├── m_cast_tiempos_advnav.html  # Advento e Natal
    ├── m_cast_tiempos_cuaresma.html # Quaresma
    ├── m_cast_tiempos_pascua.html  # Páscoa
    ├── m_cast_tiempos_ordinario.html # Tempo Ordinário
    └── m_cast_tiempos_semanasta*.html # Semana Santa
├── lecturas/                      # Leituras (~90 arquivos)
├── santos/                        # Santos por mês (12+ arquivos)
├── plegarias_euc/                 # Orações Eucarísticas
├── prefacios/                     # Prefácios
└── comunes_votivas/               # Missas comuns e votivas
```

7. Sistema de Carregamento Dinâmico

O conteúdo é carregado dinamicamente via AJAX:

7.1. Função `carga_pagina()`

```
function carga_pagina(url) {
    return new Promise((resolve, reject) => {
        $.ajax({
            url: url,
            dataType: 'html',
            success: function(data) {
                resolve(data);
            },
            error: function(xhr, status, error) {
                reject(error);
            }
        });
    });
}
```

7.2. Exemplo de Uso

```
// Carrega o Ordinário da Missa
carga_pagina("m_estructura/ordinario/m_estructura_ordinario.html")
    .then((contenido) => {
        $("#tab_o").html(contenido);
        arreglaCarga("o");
    });
});
```

7.3. Ajuste de Idiomas

Após carregar o conteúdo, os idiomas são ajustados:

```
function ajusta_idiomas(idiomal, idioma2) {
    // Mostra o idioma principal
    $("." + idiomal).css("display", "block");

    // Oculta ou mostra o segundo idioma
    if (idiomal !== idioma2) {
        $("." + idioma2).css("display", "block");
    } else {
        $("." + idioma2).css("display", "none");
    }
}
```

8. Estrutura do Conteúdo HTML

8.1. Sistema de Classes CSS

O conteúdo utiliza um sistema de classes para identificar elementos:

```
<!-- Estrutura no template (m_estructura) -->
<div class="padre padre_1">
    <!-- Container para inserir conteúdo -->
</div>

<!-- Conteúdo no idioma específico (m_cast) -->
<div class="cast hijo hijo_1" ontouchend="cambialengua(this)">
    <h1>ORDINARIO DE LA MISA</h1>
</div>
```

8.2. Elementos de Interação

```
<!-- Botão para expandir/colapsar seções -->
<div class="boton_mas">
    <span class="boton" ontouchend="muestraono('ritos_iniciais', true);">
        +
    </span>
</div>

<!-- Botão de opções litúrgicas -->
<div class="div-botones">
    <div class="bot_ord" ontouchend="activa_bot(this,'ord','global');">
        Ord
    </div>
    <div class="bot_tmp" ontouchend="activa_bot(this,'tmp','global');">
        Tmp
    </div>
</div>
```

8.3. Rubricas e Formatação

As rubricas (instruções em vermelho) são identificadas por classes:

```
<!-- Texto vermelho (rubrica) -->
<span class="red">El sacerdote dice:</span>

<!-- Texto do celebrante -->
<span class="cap">E</span>n el nombre del Padre...

<!-- Cruz vermelha -->
<span class="cruzroja">⌘</span>
```

9. Arquivo `mis_funciones_misal.js`

Este arquivo (3691 linhas) contém as funções principais:

9.1. Funções de Preferências

```
// Lê preferência do localStorage
function dime_pref(key, defecto) {
    var resultado = window.localStorage.getItem(key);
    if (resultado == null) resultado = defecto;
    return resultado;
}

// Salva preferência com tratamento de erro
function pon_pref(key, value) {
    try {
        localStorage.setItem(key, value);
    } catch (e) {
        limpiarLocalStorageInteligente();
        try {
            localStorage.setItem(key, value);
        } catch (e2) {
            alert("Erro ao salvar preferência");
        }
    }
}
```

9.2. Inicialização do iScroll

```
myScroll = new iScroll("contenedor", {
    hScrollbar: false,
    vScrollbar: false,
    bounce: false,
    hideScrollbar: true,
    momentum: true,
    onBeforeScrollMove: function(e) {
        // Detecta swipe horizontal para "virar página"
        if (this.distX > 40 && this.absDistX > this.absDistY) {
            myScroll.scrollTo(0, -paginaAltura, 300, true);
        }
    },
    onScrollEnd: function() {
        // Salva posição do scroll
        pon_pref("scroll_" + mipestana + puntero, myScroll.y);
    }
});
```

9.3. Sistema de Navegação por Abas

```
function toggle_pestanas(indice) {
    // Oculta todas as abas
    $('[id^="tab_"]').css("display", "none");

    // Mostra a aba selecionada
    var abas = ['o', 't', 's', 'pf', 'pe', 'l'];
    var aba = abas[indice];
    $("#tab_" + aba).css("display", "block");

    // Atualiza indicador visual
    $(".navbar a").removeClass("active");
    $(".navbar a").eq(indice).addClass("active");
}
```

10. Arquivo `mis_funciones_devoc.js`

Este arquivo (543 linhas) contém funções auxiliares e do devocionário:

10.1. Função de Troca de Idioma

```
function cambialengua(objeto) {
    // Alterna entre idioma principal e secundário ao tocar
    if (objeto.className == mimisal_2) {
        // Mostra idioma 1, oculta idioma 2
        $(padre).children("." + mimisal_1).css("display", "block");
        $(padre).children("." + mimisal_2).css("display", "none");
    } else if (objeto.className == mimisal_1) {
        // Mostra idioma 2, oculta idioma 1
        $(padre).children("." + mimisal_2).css("display", "block");
        $(padre).children("." + mimisal_1).css("display", "none");
    }
}
```

10.2. Ajuste de Margens de Segurança (iOS)

```
function getSafeAreaTop() {
    let testDiv = document.createElement("div");
    testDiv.style.cssText =
        "position: absolute; top: 0; " +
        "height: env(safe-area-inset-top); visibility: hidden;";
    document.body.appendChild(testDiv);
    let safeAreaTop = testDiv.offsetHeight;
    document.body.removeChild(testDiv);
    return safeAreaTop || 0;
}

function arregla_top() {
    let safeAreaTop = getSafeAreaTop();
    $('#header').css('padding-top',
        (parseInt(dime_pref('margen_superior_defecto', 0)) + safeAreaTop) + 'px'
    );
    $("#contenedor").css("top", ($("#header").outerHeight(true) + "px"));
}
}
```

10.3. Função de Scroll para Elemento

```
function scrollToElementWithOffset(element, offsetEm = 0) {
    element.scrollIntoView({
        behavior: "smooth",
        block: "start"
    });

    const parent = document.getElementById('contenedor');
    const fontSize = parseFloat(getComputedStyle(element).fontSize);
    const offset = -offsetEm * fontSize;

    // Aplica offset adicional após o scroll terminar
    setTimeout(() => {
        parent.scrollBy({ top: offset, behavior: "smooth" });
    }, 100);
}
```

11. Arquivo CSS: `misal.css`

O arquivo de estilos (1311 linhas) define toda a aparência:

11.1. Classes de Formatação Litúrgica

```
/* Texto em vermelho (rubricas) */
.red, .re { color: red; }

/* Negrito vermelho */
.rebo { color: red; font-weight: bold; }

/* Centralizado vermelho */
.rece { color: red; text-align: center; }

/* Cruz vermelha */
.cruzroja { color: red; font-size: 1.2em; }

/* Versaletes (inicial maiúscula) */
.cap { font-variant: small-caps; font-size: 1.2em; }

/* Texto do povo (respostas) */
.pueblo { font-weight: bold; }
```

11.2. Classes de Layout

```
/* Container principal */
#contenedor {
    position: absolute;
    top: 2.1em;
    bottom: -2em;
    left: 0;
    width: 100vw;
    overflow: auto;
}

/* Scroller interno */
#scroller {
    flex: 1;
    padding-top: 1em;
    touch-action: pan-y;
}

/* Texto incluído (citações) */
.texto_incluido {
    background-color: whitesmoke;
    padding: 0.5em;
    border-left: 3px solid maroon;
}
```

11.3. Modo Noturno

```
.noche {
    background-color: #121212;
    color: white;
}

.noche .texto_incluido {
    background-color: DarkSlateGray;
}
```

12. Sistema de Preferências

O app armazena diversas preferências no localStorage:

Chave	Descrição	Padrão
valor_noche	Modo noturno (0/1)	0
mimisal_1	Idioma principal	cast
mimisal_2	Idioma secundário	latin
tamano_letra	Tamanho da fonte (%)	100
margin_superior_defecto	Margem superior (px)	0
margin_inferior_defecto	Margem inferior (px)	0
pasarpagina_defecto	Virar página com swipe	1
misintenciones	Intenções pessoais	---
scroll_o*	Posição de scroll por aba	0

12.1. Página de Preferências

O arquivo `preferencias.html` permite configurar:

- Idioma principal e secundário
- Tamanho da fonte
- Modo noturno
- Margens de segurança
- País/região litúrgica (Argentina, Brasil, etc.)
- Configurações de exibição de texto

13. Plugins Cordova

13.1. cordova-plugin-device

Fornece informações sobre o dispositivo:

```
// Após deviceready
console.log(device.platform);      // 'Android', 'iOS', 'Browser'
console.log(device.version);       // '10.0'
console.log(device.manufacturer);  // 'Samsung'
console.log(device.model);         // 'SM-G950F'
```

13.2. cordova-plugin-insomnia

Mantém a tela ligada durante a leitura:

```
window.plugins.insomnia.keepAwake();
// ou
window.plugins.insomnia.allowSleepAgain();
```

13.3. cordova-plugin-file

Acesso ao sistema de arquivos (para cache offline):

```
window.requestFileSystem(LocalFileSystem.PERSISTENT, 0,
  function(fs) {
    // Acesso ao sistema de arquivos
  }
);
```

13.4. cordova-plugin-globalization

Internacionalização:

```
navigator.globalization.getLocaleName(
  function(locale) {
    console.log('locale: ' + locale.value);
  }
);
```

14. Biblioteca JSCalendar

O app utiliza uma versão customizada do JSCalendar para exibir o calendário litúrgico:

14.1. Arquivos Principais

```
jscalendar/
├── calendar.js          # Lógica do calendário
├── calendar-setup.js    # Inicialização
├── calendar-brown.css   # Tema marrom (usado)
└── lang/
    ├── calendar-lat.js  # Latim (padrão)
    ├── calendar-es.js   # Espanhol
    ├── calendar-pt.js   # Português
    └── ...
└── skins/
    └── aqua/            # Tema alternativo
```

14.2. Inicialização

```
Calendar.setup({
    inputField: "fecha",
    button: "trigger",
    ifFormat: "%d.%m.%Y",
    daFormat: "%Y/%m/%d",
    onSelect: function(calendar, date) {
        // Callback quando data é selecionada
        actualizaFecha(date);
    }
});
```

15. Tratamento de Erros

O app implementa um sistema global de captura de erros:

```
function initGlobalErrorHandler() {
    let osInfo = 'SO desconocido';
    let deviceInfo = 'Dispositivo desconocido';

    document.addEventListener('deviceready', function() {
        if (window.device) {
            osInfo = `${device.platform} ${device.version}`;
            deviceInfo = `${device.manufacturer} ${device.model}`;
        }
    });

    // Captura erros JavaScript
    window.addEventListener('error', function(event) {
        const fullMessage = [
            '🔴 Error de JavaScript',
            `SO: ${osInfo}`,
            `Dispositivo: ${deviceInfo}`,
            `Mensaje: ${event.message}`,
            `Archivo: ${event.filename}`,
            `Línea: ${event.lineno}`
        ].join('\n');

        console.error(fullMessage);
        alert(fullMessage);
    });

    // Captura erros em Promises
    window.addEventListener('unhandledrejection', function(event) {
        console.error('Promise rejeitada:', event.reason);
        alert('Erro em promise: ' + event.reason);
    });
}
```

16. Fluxo de Exibição do Conteúdo

Quando o usuário seleciona uma data, o seguinte fluxo ocorre:

1. Seleção da Data

```
Calendar.onSelect() → actualizaFecha(date)
```

2. Cálculo Litúrgico

Determina: tempo litúrgico, ciclo, santos do dia

3. Montagem dos Links

```
mienlace_misa["dia"] = "m_estructura/tiempos/...#A123"  
mienlace_lecturas["dia"] = "m_estructura/lecturas/...#A123"
```

4. Carregamento Dinâmico

```
carga_página(mienlace_misa["dia"]).then(....)
```

5. Ajuste de Idiomas

```
ajusta_idiomas(mimisal_1, mimisal_2)
```

6. Refresh do Scroll

```
myScroll.refresh()
```

Listagem 2: Fluxo de carregamento de conteúdo

17. Resumo da Arquitetura

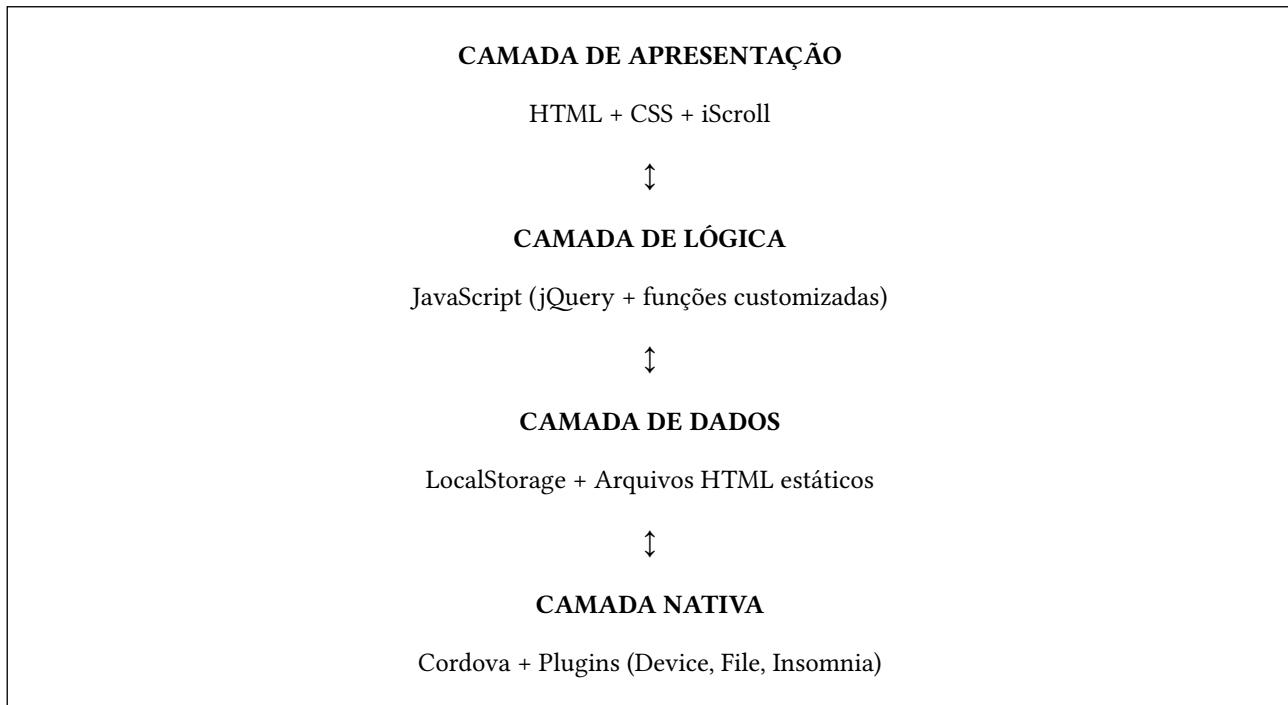


Figura 1: Arquitetura em camadas do aplicativo

17.1. Pontos Fortes

- **Conteúdo offline**: Todo o missal está embutido no app
- **Multi-idioma**: 8 idiomas suportados
- **Calendário completo**: Cálculo automático de todas as festas
- **Personalização**: Diversas preferências do usuário
- **Compatibilidade**: Funciona em mobile e navegador

17.2. Pontos de Atenção

- **Tamanho**: Aproximadamente 63MB de conteúdo HTML
- **Manutenção**: Código legado com muita duplicação
- **Performance**: Carregamento síncrono em algumas partes
- **LocalStorage**: Limite de 5MB pode ser atingido

Documentação gerada em dezembro de 2025