



# Guía de Instalación - App Yacht v2.0.0

## Requisitos del Sistema

### WordPress

- **Versión mínima:** WordPress 5.0+
- **Versión recomendada:** WordPress 6.0+
- **PHP:** 7.4+ (Recomendado: PHP 8.0+)
- **MySQL:** 5.7+ o MariaDB 10.2+

### Servidor

- **Memoria:** Mínimo 256MB, Recomendado 512MB+
- **Almacenamiento:** 50MB espacio libre
- **Extensiones PHP requeridas:**
  - `curl` (para scraping de yates)
  - `json` (procesamiento de datos)
  - `mbstring` (manejo de strings)
  - `libxml` (parsing HTML)

### Configuración de PHP

```
max_execution_time = 60
memory_limit = 256M
upload_max_filesize = 10M
post_max_size = 10M
```



## Instalación Nueva

### Paso 1: Preparación

```
# Crear backup del tema actual
cp -r /path/to/theme /path/to/backup/theme_backup_$(date +%Y%m%d)

# Verificar permisos
find /path/to/theme -type f -exec chmod 644 {} \;
find /path/to/theme -type d -exec chmod 755 {} \;
```

### Paso 2: Despliegue de Archivos

Archivos Nuevos a Copiar:

```

app_yacht/
├── core/
│   ├── bootstrap.php      ← NUEVO
│   ├── container.php      ← NUEVO
│   ├── config.php         ← NUEVO
│   └── app-yacht.php      ← ACTUALIZADO
├── shared/
│   ├── interfaces/       ← NUEVO
│   └── helpers/          ← NUEVO
└── modules/
    ├── yachtinfo/        ← NUEVO
    ├── calc/calc-service.php ← NUEVO
    ├── render/           ← NUEVO
    └── mail/mail-service.php ← NUEVO

```

### Paso 3: Actualizar functions.php

```

// Reemplazar la línea existente:
// require_once get_template_directory() . '/app_yacht/core/yacht-functions.php';

// Por:
//===== APP YACHT v2.0 =====//
require_once get_template_directory() . '/app_yacht/core/bootstrap.php';

add_action('init', function() {
    if (!class_exists('AppYachtBootstrap')) {
        error_log('Error: AppYachtBootstrap class not found');
        return;
    }

    try {
        AppYachtBootstrap::init();
    } catch (Exception $e) {
        error_log('Error inicializando App Yacht: ' . $e->getMessage());
    }
});

// Mantener compatibilidad
$legacy_yacht_functions = get_template_directory() . '/app_yacht/core/yacht-functions.php';
if (file_exists($legacy_yacht_functions)) {
    require_once $legacy_yacht_functions;
}
//===== APP YACHT v2.0 END=====//

```

## Paso 4: Verificación de Instalación

### Test Básico

```
// Agregar temporalmente al final de functions.php para testing
add_action('init', function() {
    if (is_admin() && current_user_can('administrator')) {
        $health = app_yacht_installation_check();
        if (!$health['success']) {
            add_action('admin_notices', function() use ($health) {
                echo '<div class="notice notice-error"><p>App Yacht: ' .
$health['message'] . '</p></div>';
            });
        }
    }
});

function app_yacht_installation_check() {
    // Verificar clases principales
    if (!class_exists('AppYachtBootstrap')) {
        return ['success' => false, 'message' => 'Bootstrap class not found'];
    }

    if (!class_exists('AppYachtContainer')) {
        return ['success' => false, 'message' => 'Container class not found'];
    }

    if (!class_exists('AppYachtConfig')) {
        return ['success' => false, 'message' => 'Config class not found'];
    }

    // Verificar servicios
    try {
        $container = AppYachtBootstrap::getContainer();
        $services = $container->getRegisteredServices();

        $requiredServices = ['yacht_info_service', 'calc_service', 'render_engine',
'mail_service'];
        foreach ($requiredServices as $service) {
            if (!in_array($service, $services)) {
                return ['success' => false, 'message' => "Service '$service' not re-
gistered"];
            }
        }
    } catch (Exception $e) {
        return ['success' => false, 'message' => 'Error initializing: ' . $e->getMes-
sage()];
    }

    return ['success' => true, 'message' => 'Installation successful'];
}
```

### Test de Funcionalidad

1. Ir a la página de App Yacht
2. Verificar que carga sin errores
3. Realizar un cálculo básico
4. Generar un template
5. Verificar logs de error

## Configuración Post-Instalación

### Configurar Dominios de Scraping

En `core/config.php`, personalizar dominios permitidos:

```
'scraping' => [
    'allowed_domains' => [
        'charterworld.com',
        'yachtcharterfleet.com',
        'tu-dominio-personalizado.com'
    ]
]
```

### Configurar Caché

```
'cache' => [
    'enabled' => true,
    'default_duration' => 3600, // 1 hora
    'cleanup_interval' => 24 * HOUR_IN_SECONDS
]
```

### Configurar Email

```
'mail' => [
    'default_sender' => 'tu-email@ejemplo.com',
    'outlook_enabled' => true,
    'max_recipients' => 50
]
```

## Troubleshooting de Instalación

### Error: “Class not found”

**Causa:** Archivo no cargado o ruta incorrecta

**Solución:**

```
# Verificar que los archivos existen
ls -la app_yacht/core/bootstrap.php
ls -la app_yacht/core/container.php

# Verificar permisos
chmod 644 app_yacht/core/*.php
```

### Error: “Service not registered”

**Causa:** Servicio no se registró correctamente

**Solución:**

```
// Verificar en bootstrap.php que todos los servicios están registrados
$container->register('service_name', function() {
    return new ServiceClass();
});
```

## Error: Memory limit

**Causa:** Límite de memoria insuficiente

**Solución:**

```
// En wp-config.php
ini_set('memory_limit', '512M');

// 0 en .htaccess
php_value memory_limit 512M
```

## Error: Timeout en scraping

**Causa:** Configuración de timeout muy baja

**Solución:**

```
// En config.php
'scraping' => [
    'timeout' => 60, // Aumentar timeout
    'max_redirects' => 5
]
```



## Verificación de Performance

### Herramientas de Monitoreo

```
// Agregar en functions.php para monitoreo temporal
add_action('wp_footer', function() {
    if (current_user_can('administrator') && isset($_GET['debug'])) {
        $memory = memory_get_peak_usage(true) / 1024 / 1024;
        $time = microtime(true) - $_SERVER['REQUEST_TIME_FLOAT'];
        echo "<!-- Memory: {$memory}MB, Time: {$time}s -->";
    }
});
```

### Benchmarks Esperados

- **Tiempo de carga inicial:** < 50ms adicionales
- **Memoria adicional:** < 1MB
- **Tiempo de cálculo:** < 100ms
- **Tiempo de generación de template:** < 200ms



## Actualización desde Versión Anterior

### Checklist Pre-Actualización

- ☐ ☒ Backup completo creado

- [ ] ☒ Entorno de desarrollo disponible
- [ ] ☒ Verificación de requisitos del sistema
- [ ] ☒ Lista de customizaciones existentes
- [ ] ☒ Plan de rollback preparado

## Proceso de Actualización

### 1. Modo Mantenimiento:

```
// En wp-config.php
define('WP_MAINTENANCE_MODE', true);
```

### 1. Aplicar Archivos:

```
# Copiar archivos nuevos
cp -r nueva_version/app_yacht/* tema_actual/app_yacht/

# Actualizar functions.php
# (seguir paso 3 de instalación nueva)
```

### 1. Verificar Funcionamiento:

```
# Verificar logs
tail -f /path/to/wordpress/debug.log

# Test básico
curl -I http://tu-sitio.com/app-yacht/
```

### 1. Desactivar Mantenimiento:

```
// Comentar en wp-config.php
// define('WP_MAINTENANCE_MODE', true);
```

## Consideraciones de Seguridad

### Permisos de Archivos

```
# Archivos PHP
find app_yacht -name "*.php" -exec chmod 644 {} \;

# Directorios
find app_yacht -type d -exec chmod 755 {} \;

# No ejecutables
find app_yacht -name "*.css" -o -name "*.js" -exec chmod 644 {} \;
```

### Variables de Entorno

```
// En wp-config.php - configuraciones sensibles
define('APP_YACHT_DEBUG', false);
define('APP_YACHT_CACHE_ENABLED', true);
```

## Firewall y Rate Limiting

```
# En .htaccess
<Limit POST>
    Require valid-user
</Limit>
```

## Soporte de Instalación

### Auto-Diagnóstico

```
// Ejecutar en WordPress admin
function app_yacht_diagnostics() {
    $report = [];

    // PHP Version
    $report['php_version'] = PHP_VERSION;
    $report['php_requirements'] = version_compare(PHP_VERSION, '7.4', '>=');

    // WordPress Version
    $report['wp_version'] = get_bloginfo('version');
    $report['wp_requirements'] = version_compare(get_bloginfo('version'), '5.0', '>=');

    // Extensions
    $required_extensions = ['curl', 'json', 'mbstring', 'libxml'];
    foreach ($required_extensions as $ext) {
        $report['extensions'][$ext] = extension_loaded($ext);
    }

    // App Yacht Classes
    $required_classes = ['AppYachtBootstrap', 'AppYachtContainer', 'AppYachtConfig'];
    foreach ($required_classes as $class) {
        $report['classes'][$class] = class_exists($class);
    }

    // Services
    try {
        $container = AppYachtBootstrap::getContainer();
        $report['services'] = $container->getRegisteredServices();
    } catch (Exception $e) {
        $report['services_error'] = $e->getMessage();
    }

    return $report;
}
```

## Información para Soporte

Si necesitas ayuda, incluye:

1. **Diagnóstico completo** (función anterior)
2. **Logs de error** relevantes
3. **Configuración del servidor**
4. **Pasos para reproducir el problema**
5. **Versión anterior funcionando**

**¡Instalación completada! App Yacht v2.0.0 está listo para usar con arquitectura mejorada y máxima compatibilidad.**