Nuoto Libero Le Naiadi - Guida Builder Free + Aruba

Generato il: 2026-02-19 19:33

```
# Guida Integrazione Builder Free + Aruba + Dashboard Admin/Ufficio

Data guida: 19 febbraio 2026
Progetto: Nuoto Libero Le Naiadi

## Obiettivo reale
Usare Builder.io versione free solo come CMS visuale, mantenendo il sito online su Aruba (dominio
proprietario) con database MySQL Aruba e pannelli admin/ufficio gia presenti nel progetto.

Builder non ospita il sito.
Hosting e dominio restano su Aruba.

---

## Architettura finale (chiara)
1. Frontend e backend PHP online su Aruba.
2. DB MySQL su Aruba (gestito da phpMyAdmin Aruba).
3. Builder.io free usato per contenuti (model `page`), letti dal backend via API.
4. Admin e ufficio gestiscono CMS da `piscina-php/dashboard-cms-builder.php`.

---

## Fase 0 - Verifica push/Git prima di tutto

### Caso A - Cartella con Git presente
Esegui da terminale nella root progetto:

```powershell
git status --short --branch
git fetch origin
git checkout main
git pull --rebase origin main
git add -A
git commit -m "chore: aggiornamenti pre deploy aruba builder"
git push origin main
git log -1 --oneline
```

Caso B - Cartella senza `.git` (caso attuale rilevato)
Se `git status` risponde `fatal: not a git repository`, in questa cartella non puoi fare push.

Procedura corretta:
1. Fai una copia di sicurezza della cartella corrente.
2. Clona il repo ufficiale in una cartella nuova:
   `git clone <URL_REPO> Nuoto-Libero-Le-Naiadi-clean`
3. Copia dentro la nuova cartella solo le modifiche necessarie (file progetto, non backup inutili).
4. Esegui commit e push dalla cartella clonata.
5. Solo dopo il push procedi con il deploy Aruba.

Fase 1 - Backup obbligatorio pre deploy
1. Backup file completo (zip del progetto).
2. Backup DB locale `nuoto_libero` in SQL.
3. Salva entrambi in cartella esterna (es. `C:\xampp\htdocs\_backups\`).

Fase 2 - Setup Builder.io FREE (senza hosting Builder)
1. Accedi a Builder.io con account free.
2. Crea o apri lo Space del progetto.
3. Crea model contenuto `page`.
4. Crea almeno 1 entry pubblicata:
   - `urlPath: /`
   - titolo e blocchi base
5. Recupera la Public API Key dallo Space.

Nota: in questo progetto la chiave e letta da `BUILDER_API_KEY` (oppure fallback
`BUILDER_PUBLIC_API_KEY`).
```

Fase 3 - Configurazione progetto (locale e poi Aruba)
Aggiorna `.env`:

```env
APP_ENV=production
APP_BASE_URL=https://tuodominio.it
SITE_MODE=full

DB_HOST=...
DB_PORT=3306
DB_NAME=...
DB_USER=...
DB_PASS=...

JWT_SECRET=STRINGA_LUNGA_E_UNICA

MAIL_ENABLED=true
MAIL_SMTP_HOST=...
MAIL_SMTP_PORT=587
MAIL_SMTP_USER=...
MAIL_SMTP_PASS=...
MAIL_FROM_EMAIL=info@tuodominio.it
MAIL_ADMIN_EMAIL=info@tuodominio.it

BUILDER_API_KEY=LA_TUA_PUBLIC_API_KEY
BUILDER_PUBLIC_API_KEY=LA_TUA_PUBLIC_API_KEY
BUILDER_CDN_BASE_URL=https://cdn.builder.io
BUILDER_WEBHOOK_SECRET=metti_un_secret_solo_se_userai_webhook
```

Fase 4 - Integrazione Admin e Ufficio (gia pronta nel codice)
La pagina CMS e:
- `piscina-php/dashboard-cms-builder.php`

Accesso consentito ai ruoli:
- `admin`
- `ufficio`
- `segreteria`

Test rapido:
1. Login come admin o ufficio.
2. Apri dashboard CMS Builder.
3. Premi "Test Builder.io".
4. Verifica risposta:
   - se key mancante: messaggio configurazione assente
   - se key corretta: contenuto pubblicato trovato (o messaggio "nessun contenuto" sul path)

Endpoint tecnico:
- `GET /api/cms.php?action=builder-stub&model=page&url_path=/`

Webhook opzionale:
- `POST /api/cms.php?action=builder-webhook`
- Header opzionale: `X-Builder-Secret: <BUILDER_WEBHOOK_SECRET>`

Fase 5 - Creazione DB su Aruba (phpMyAdmin Aruba)
1. Nel pannello Aruba crea database MySQL.
2. Prendi i dati connessione:
   - host
   - nome database
   - username
   - password
3. Apri phpMyAdmin Aruba.
4. Importa SQL base:
   - `db/CREATE_DATABASE_FROM_ZERO.sql`
5. Importa eventuali migrazioni mancanti (in ordine data).
6. Verifica tabelle chiave:
   - `profili`

   - `ruoli`
   - `acquisti`
   - `check_ins`
   - `impostazioni_operative`
   - tabelle CMS (`cms_*`)

Fase 6 - Upload su Aruba (hosting vero)
1. Carica file via FTPS (consigliato FileZilla) nella root pubblica Aruba.
2. Verifica presenza file chiave:
   - `.htaccess`
   - `index.php`
   - cartelle `api`, `piscina-php`, `assets`, `vendor`
3. Carica `.env` produzione valorizzato.
4. Permessi consigliati:
   - file 644
   - cartelle 755
   - cartelle scrivibili (`logs`, `uploads`) 755/775 in base al piano Aruba

---

Fase 7 - Test post deploy (ordine consigliato)
1. URL pubblici:
   - `/`
   - `/landing.php`
   - `/login.php`
2. Login ruoli:
   - admin
   - ufficio
   - bagnino
   - utente
3. Dashboard CMS:
   - apri `piscina-php/dashboard-cms-builder.php`
   - test Builder da interfaccia
4. API CMS:
   - `GET /api/cms.php?action=builder-stub&model=page&url_path=/`
5. Flussi business:
   - richiesta utente
   - approvazione admin/ufficio
   - QR e check-in
6. Form contatti + email

Se un test critico fallisce, non aprire ufficialmente il servizio.

Fase 8 - Go live e sicurezza minima
1. Conferma SSL attivo sul dominio Aruba.
2. Tieni `SITE_MODE=full` solo quando test completati.
3. Mantieni backup giornaliero DB e settimanale file.
4. Non salvare segreti in chiaro nei file versionati.

Fase 9 - Rollback rapido (piano emergenza)
1. Ripristina zip file precedente.
2. Reimporta SQL backup precedente.
3. Ripristina `.env` precedente.
4. Testa immediatamente home + login + dashboard admin.

Tempo target rollback: 15-20 minuti.

Checklist finale sintetica
- [ ] Push confermato su repo `main` (da cartella con `.git`)
- [ ] Backup file + DB completato
- [ ] DB Aruba creato e importato correttamente
- [ ] `.env` produzione configurato (DB, JWT, SMTP, Builder key)
- [ ] Upload file completato su Aruba
- [ ] Login ruoli ok
- [ ] Dashboard CMS accessibile da admin/ufficio

- [ ] Test Builder stub ok
- [ ] Flussi QR/check-in e contatti ok
- [ ] Piano rollback pronto

Nota importante per il tuo caso
Nel workspace corrente (19 febbraio 2026) e stato rilevato che manca `.git`.
Quindi il push non puo essere eseguito da questa cartella finche non lavori su una copia clonata correttamente dal repository remoto.