

# Branch Management Guide

---

## Current State

The repository currently has the following branches:

- **main** (commit: 8272cbd) - Contains only the initial LICENSE commit
- **master** (commit: 5d48d18) - Contains the complete application with 14 commits including full project structure
- **copilot/fix-8ff01a8d-4999-4d5d-a3ff-a50bc427fa0c** (current working branch)

## Key Differences

### Main branch contains:

- Only LICENSE file
- Single initial commit

### Master branch contains:

- Complete application structure (frontend, backend, configs)
- Package.json with dependencies
- CI/CD workflows
- Linting and formatting configurations
- Full project history (14 commits)
- All application code and features

## Required Changes

1. Delete the **main** branch
2. Keep only the **master** branch
3. Rename **master** branch to **main**

## GitHub Operations Required

Since this environment has limitations on direct Git operations affecting remote branches, the following operations need to be performed directly in GitHub:

### Step 1: Delete the **main** branch

This can be done via GitHub web interface or using GitHub CLI with appropriate permissions:

#### Via GitHub Web Interface:

1. Go to the repository: <https://github.com/asdrubalfuentes/cotizador>
2. Click on "branches" (should show 3 branches)
3. Find the **main** branch
4. Click the delete button (trash icon) next to **main** branch

### Via GitHub CLI (if you have access):

```
gh api repos/asdrubalfuentes/cotizador/git/refs/heads/main -X DELETE
```

### Step 2: Rename **master** to **main**

This can be done via GitHub web interface:

#### Via GitHub Web Interface:

1. Go to repository Settings
2. Scroll down to "Default branch" section
3. Click on the branch name (currently might be **master**)
4. Select or type **main** as the new default branch name
5. GitHub will offer to rename the branch - accept this option

#### Alternative via GitHub CLI:

```
# Rename the branch
gh api repos/asdrubalfuentes/cotizador/branches/master/rename -f new_name=main
```

### Step 3: Verify Changes

After completing the above steps:

1. The **main** branch should no longer exist
2. The **master** branch should be renamed to **main**
3. The new **main** branch should be set as the default branch

## Local Repository Updates

After the remote changes are made, local repositories need to be updated:

```
# Fetch latest changes
git fetch origin

# If you have a local main branch, delete it
git branch -D main

# If you have a local master branch, delete it
git branch -D master

# Create new main branch tracking the renamed remote branch
git checkout -b main origin/main
```

```
# Set upstream  
git branch --set-upstream-to=origin/main main
```

## Important Notes

- **CRITICAL:** The **master** branch contains the actual application with 14 commits, while **main** only has the LICENSE file
- **Data Safety:** Ensure all important commits from both branches are preserved (master has the complete application)
- The **master** branch should become the new **main** branch as it contains the working application
- Consider creating a backup before making these changes
- Update any CI/CD configurations that reference the old branch names (note that CI workflow already supports both **master** and **main**)
- Notify team members about the branch name changes
- The existing CI workflow in [.github/workflows/ci.yml](#) already handles both **master** and **main** branches