Here are the key steps for Thiago and Gerson to integrate their remaining assignments into GitHub:

**Steps for Thiago and Gerson to Integrate Their Work:**

1. **Pull the Latest main Branch:**
   * Before starting any new work or integrating, both Thiago and Gerson should ensure their local main branch is up-to-date with your latest changes.
   * git checkout main
   * git pull origin main
2. **Create/Switch to Their Feature Branches:**
   * They should then switch to their dedicated feature branches (e.g., thiago/interpolation-search, gerson/ui-integration). If they haven't created them yet, they should do so now based on the updated main.
   * git checkout -b thiago/interpolation-search (if creating new)
   * git checkout thiago/interpolation-search (if already exists)
   * *(Repeat for Gerson's branch)*
3. **Implement Their Assigned Features:**
   * **Thiago:**
     + Implement the interpolationSearch function within the ProjectUtils namespace in ProjectUtils.h and its corresponding .cpp file (if you decide to separate it, otherwise directly in ProjectUtils.h).
     + Implement the search query logic and result verification for Interpolation Search within main.cpp, calling his interpolationSearch and measureSearchTime.
     + Ensure his code includes appropriate comments and handles edge cases.
   * **Gerson:**
     + Continue working on main.cpp to build out the full command-line user interface.
     + Integrate calls to your generateAndSortDataset, loadAndSortDatasetFromFile, and jumpSearch functions.
     + Integrate calls to Thiago's interpolationSearch (once available).
     + Implement robust input validation for all user prompts.
     + Ensure clear and formatted output for all operations, including search results and timings.
4. **Commit Their Changes Regularly:**
   * As they make progress, they should commit their changes frequently with descriptive messages.
   * git add .
   * git commit -m "Implemented Interpolation Search algorithm"
5. **Push Their Feature Branches to GitHub:**
   * They should regularly push their local feature branches to the remote GitHub repository.
   * git push origin thiago/interpolation-search
6. **Create Pull Requests (PRs):**
   * Once their respective features are complete and they've tested them on their local machines, they should create a Pull Request on GitHub.
   * The PR should be from their feature branch (e.g., thiago/interpolation-search) *into* the main branch.
   * They should provide a clear description of the changes made and link to any relevant issues or tasks.
7. **Review and Merge:**
   * As a team, you should review each other's Pull Requests. This is a crucial step for code quality, consistency, and catching potential bugs.
   * Once reviewed and approved, the PR can be merged into the main branch.
8. **Keep main Clean and Functional:**
   * The main branch should always represent the most stable, working version of the project. Avoid direct commits to main for new features; always use feature branches and PRs.

By following these steps, you'll ensure a smooth collaborative process and a well-organized GitHub repository for your final submission. Good luck with the remaining work!