Simone Testino

I build SQL databases and AI-assisted tools in Python, and study how proof assistants can verify the rules that software implements.

CONTACT

Contact Simone

Professional projects

WORKING LANGUAGES

Italian Native

English C1

German C1

PROFILES & WEBSITES

GitHub

LinkedIn

simonetestino.pages.dev

01 Software systems 02 Applied AI 03 Education

Software in operation

ColivingLiguria SRL · Founder and developer · Company incorporated March 2026

I began developing the venture in June 2025 by preparing my application for Liguria's ERDF start-up grant (Action 1.3.3). I design relational schemas, APIs and interfaces for the people who manage stays, partnerships and accounts. In three systems that I build and maintain, each record has a stable identifier, a source and a history.

Living Register

Links people, stays and spaces. I model the records an operator needs to check availability and agreed steps.

Business Register

Links organisations, proposals and decisions. I track revisions so an operator can trace the source of a change.

Financial Register

Links ledger entries and cash-flow forecasts. I reconcile amounts in minor currency units with their sources.

> Working stack: Python, FastAPI, SQLite/SQLCipher and TypeScript. For migrations I compare identifiers and totals, then rerun the operation to check that it creates no duplicate records.

Applied AI and engineering data

Skill BIM · From surveys to model objects

I designed an AI-assisted BIM workflow that links survey evidence to building objects. I use Python to check IFC identifiers, storey assignments and cross-record references. The public model makes the developing geometry available for inspection.

Agent design — For each agent I define its inputs, expected output and acceptance checks. When an updated source differs from a stored value, the operator can review a proposed correction.

Grant advisory and investment — I work with technical contributors on public-funding projects. I translate budget requirements into structured fields and link cost items to the documents that support them.

> A practical example: if the source or storey of an IFC object changes, a consistency check can flag the discrepancy for the technician who owns that part of the model.

Academic background

2024–present
Amsterdam

University of Amsterdam · Institute for Logic, Language and Computation

MSc in Logic

I completed the required taught coursework of the two-year programme in one academic year, including proof theory and structures for semantics. I expect to finish my thesis in July 2027.

I studied philosophy and mathematics concurrently, finishing the philosophy degree and most of the mathematics coursework in two academic years.

LMU München

BSc in Mathematics

Degree in progress. My thesis was assessed in December 2024. I studied model theory, algebra and measure theory in German.

LMU München

BA in Philosophy

Awarded July 2024, with Mathematics as a minor. My thesis studied syntactic structures in language and mathematical logic.

At Futura I taught mathematics, logic and physics, using Python and Manim to animate mathematical arguments and help students follow each step.

Verification and technical practice

Mechanising Structuralism

My MSc thesis asks when a translation between formal theories preserves what their statements mean. My private initial encoding in Isabelle/HOL covers first-order terms, formulas and translations of predicate symbols.

Document automation — I wrote Python adapters that fill LaTeX templates from approved fields. An operator can correct the source record and regenerate a document without retyping the figures.

Tests and proofs — In a database migration I can test whether identifiers survive a particular import. For a formal proof I first need to state what a valid input is and what the transformation must preserve for every such input.

Python	■■■■■■	FastAPI services, transaction reconciliation and IFC validation.
LaTeX automation	■■■■■■	Technical documents generated from structured records.
SQL	■■■■■□	Relational schemas, data provenance and database migrations.

Learning in progress · Lean 4 · Isabelle/HOL · Rust · Aeneas

In my planned six-month research project at the University of Bologna (January–July 2027), I aim to formalise proofs in Lean and Isabelle and use Aeneas to verify supported Rust code by translating it into Lean. I plan to publish examples on GitHub and my website.