Simone Testino

Ich entwickle SQL-Datenbanken und KI-gestützte Werkzeuge in Python. Mich interessiert, wie sich die im Code umgesetzten Regeln mit Beweisassistenten prüfen lassen.

KONTAKT

Kontakt zu Simone
Berufliche Projekte

ARBEITSSPRACHEN

Italienisch Muttersprache
Englisch C1
Deutsch C1

PROFILE & WEBSITES

GitHub
LinkedIn
simonetestino.pages.dev

01 Softwaresysteme 02 Angewandte KI 03 Ausbildung

Software im Einsatz

ColivingLiguria SRL · Gründer und Entwickler · Gesellschaft im März 2026 gegründet

Im Juni 2025 begann ich mit meinem Antrag auf EFRE-Gründungsförderung in Ligurien (Maßnahme 1.3.3). Für die Verwaltung von Aufenthalten, Partnerschaften und Finanzen entwickle ich relationale Schemata, APIs und Oberflächen. In meinen drei Systemen hat jeder Datensatz eine dauerhafte Kennung, eine Quelle und eine Änderungshistorie.

Living Register

Verknüpft Personen, Aufenthalte und Räume. Ich modelliere die Daten für die Prüfung von Verfügbarkeiten und vereinbarten Schritten.

Business Register

Verknüpft Organisationen, Vorschläge und Entscheidungen. Ich protokolliere Änderungen, damit ihre Herkunft nachvollziehbar bleibt.

Financial Register

Verknüpft Buchungen und Liquiditätsprognosen. Ich gleiche Beträge in der kleinsten Währungseinheit mit ihren Belegen ab.

> Technik: Python, FastAPI, SQLite/SQLCipher und TypeScript. Bei Migrationen gleiche ich Kennungen und Summen ab. Ein zweiter Durchlauf prüft, ob Duplikate entstehen.

Angewandte KI und technische Daten

Skill BIM · Von der Vermessung zum Modellobjekt

Ich entwickelte einen KI-gestützten BIM-Arbeitsablauf, der Vermessungsdaten mit Bauteilen verbindet. Mit Python prüfe ich IFC-Kennungen, Geschosszuordnung und Querverweise. Im öffentlichen Modell lässt sich die entstehende Geometrie einsehen.

Entwicklung von KI-Agenten — Für jeden Agenten lege ich Eingaben, erwartete Ergebnisse und Abnahmekriterien fest. Weicht eine neue Quelle von einem gespeicherten Wert ab, kann die zuständige Person die vorgeschlagene Änderung prüfen.

Förderberatung und Investitionen — Mit Fachleuten bearbeite ich öffentlich geförderte Projekte. Budgetvorgaben übertrage ich in strukturierte Felder und verknüpfe Kostenpositionen mit den zugehörigen Belegen.

> Ein Beispiel: Wechselt die Quelle oder das Geschoss eines IFC-Objekts, kann die Konsistenzprüfung die Abweichung der zuständigen Fachkraft anzeigen.

Akademische Ausbildung

Seit 2024
Amsterdam

University of Amsterdam · Institute for Logic, Language and Computation

MSc in Logic

Ich absolvierte die Pflichtkurse des zweijährigen Programms in einem Studienjahr, darunter Beweistheorie und semantische Strukturen. Den Abschluss der Masterarbeit plane ich für Juli 2027.

Ich studierte Philosophie und Mathematik parallel und schloss in zwei Studienjahren das Philosophiestudium und den Großteil der Mathematikmodule ab.

LMU München

BSc Mathematik

Studium noch nicht abgeschlossen. Meine Bachelorarbeit wurde im Dezember 2024 bewertet. Meine Studiengebiete waren Modelltheorie, Algebra und Maßtheorie in deutscher Sprache.

LMU München

BA Philosophie

Abschluss im Juli 2024, Nebenfach Mathematik. Meine Bachelorarbeit untersuchte syntaktische Strukturen in Sprache und mathematischer Logik.

> Bei Futura unterrichtete ich Mathematik, Logik und Physik. Mit Python und Manim animierte ich mathematische Argumente, damit Lernende jeden Schritt nachvollziehen konnten.

Verifikation und technische Praxis

Mechanising Structuralism

Wann erhält die Übersetzung zwischen formalen Theorien die Bedeutung ihrer Aussagen? Meine erste private Kodierung in Isabelle/HOL umfasst Terme, Formeln und Übersetzungen von Prädikatssymbolen der Logik erster Stufe.

Dokumentautomatisierung — Meine Python-Adapter füllen LaTeX-Vorlagen aus freigegebenen Daten. Nach einer Korrektur am Quelldatensatz lässt sich ein Dokument erstellen, ohne die Zahlen erneut einzutragen.

Tests und Beweise — Bei einer Datenbankmigration kann ich prüfen, ob Kennungen einen bestimmten Import überstehen. Für einen Beweis muss ich zunächst gültige Eingaben definieren und angeben, welche Eigenschaften bei jeder solchen Eingabe erhalten bleiben.

Python	■■■■■■	FastAPI-Dienste, Transaktionsabgleich und IFC-Prüfung.
LaTeX-Automatisierung	■■■■■■	Technische Dokumente aus strukturierten Daten.
SQL	■■■■■□	Relationale Schemata, Datenherkunft und Migrationen.

In Vertiefung · Lean 4 · Isabelle/HOL · Rust · Aeneas

Im geplanten sechsmonatigen Forschungsprojekt an der Universität Bologna (Januar–Juli 2027) möchte ich Beweise in Lean und Isabelle formalisieren und unterstützten Rust-Code mit Aeneas nach Lean übersetzen und verifizieren. Beispiele plane ich auf GitHub und meiner Website.