

**TESTNO OKRUŽENJE ZA GRAFIČKI PRIKAZ GEOPROSTORNIH PODATAKA NA OSNOVU SPARQL UPITA****TEST ENVIRONMENT FOR GRAPHICAL REPRESENTATION OF GEOSPATIAL DATA BASED ON SPARQL QUERIES**Predrag Kaljević, *Fakultet tehničkih nauka, Novi Sad***Oblast – ELEKTROTEHNIKA I RAČUNARSTVO**

**Kratak sadržaj** - U ovom radu prikazan je dizajn i implementacija proširenja SPARQL Playground aplikacije u cilju testiranja mogućnosti pretrage i grafičkog prikaza geoprostornih podataka. Aplikacija je proširena podrškom za izvršavanje GeoSPARQL upita i mogućnošću grafičkog prikaza geoprostornih podataka izraženih u WKT literal formatu.

**Ključne reči:** RDF, SPARQL, GeoSPARQL, Geospatial, Query

**Abstract** - This paper presents the design and implementation of the SPARQL Playground application extension with the goal of testing the capabilities of querying and graphical representation of geospatial data. The application is extended with support for GeoSPARQL querying and the ability to graphically represent geospatial data presented in WKT literal format.

**Keywords:** RDF, SPARQL, GeoSPARQL, Geospatial, Query

**1. UVOD**

SPARQL predstavlja protokol i upitni jezik, namenjen za vršenje upita nad RDF podacima. SPARQL Playground predstavlja višenamensku standalone web aplikaciju dizajniranu i implementiranu sa ciljem učenja, predavanja i testiranja ovog upitnog jezika. Omogućava korisniku izvršavanje i modifikovanje predefinisanih upita nad testnim podacima, kao i definisanje novih upita. Oslanja se na OpenRDF Sesame 2.8.6 kao SPARQL engine.

Ideja iza ovog rada je migracija aplikacije da se oslanja na novi SPARQL engine, RDF4J. RDF4J je nastao iz Sesame i predstavlja njegovo proširenje. Razlog migraciji je podrška za GeoSPARQL, koji Sesame ne podržava. Nakon migracije, aplikacija je proširena interaktivnom mapom (uz pomoć LeafletJS biblioteke), kao i sa mogućnostima prikaza geoprostornih podataka i vršenja GeoSPARQL upita nad njima. Na kraju su pripremljeni geoprostorni podaci za testiranje performansi novih funkcionalnosti aplikacije pri radu sa velikom količinom grafičkih elemenata.

Aplikacija je razvijena u Java Spring Boot i AngularJS. Kao razvojno okruženje, korišćen je Microsoft Visual Studio Code.

**NAPOMENA:**

**Ovaj rad proistekao je iz master rada čiji mentor je bio dr Milan Gavrić, docent.**

**2. TEORIJSKE OSNOVE****2.1. RDF**

*Resource Description Framework* (RDF) predstavlja standardni model za razmenu podataka na Web-u. To je apstraktni model podataka koji organizuje elemente podataka i standardizuje kako su oni povezani međusobno. RDF model podataka služi za predstavljanje i razmenu podataka u vidu trojki, koji se sastoje od subjekta, predikata i objekta [1]. U okviru jedne trojke, subjekat predstavlja resurs definisan preko IRI-a ili praznog čvora, predikat je veza između subjekta i objekta, a objekat je neka osobina subjekta koja može biti predstavljena kao literal ili resurs.

**2.2. CIM**

Elektroenergetske kompanije koriste veliki broj različitih formata za skladištenje svojih podataka. Iako se većina ovih podataka koristi isključivo unutar kompanije, često se javlja potreba za njihovom razmenom, kako između različitih aplikacija unutar kompanije, tako i sa drugim kompanijama.

*Common Information Model* (CIM) je nastao usled potrebe informacionih sistema (koji učestvuju u upravljanju elektroenergetskim sistemima) za postojanjem opšteprihvaćenog standarda modelovanja i predstavljanja elektroenergetskih mreža. Pre nastanka CIM-a, razmena podataka između aplikacija je često zahtevala njihovu konverziju u model podataka koji je pogodan za obradu nekoj određenoj aplikaciji. Kako bi se eliminisala potreba za različitim konverzijama podataka prilikom razmene, nastao je CIM kao apstraktni model, nezavisan od platforme, za definisanje elemenata elektroenergetskog sistema.

**2.3. SPARQL**

*SPARQL Protocol and RDF Query Language* (SPARQL) je protokol i RDF upitni jezik. Dizajniran je za vršenje upita nad RDF podacima, ali nije time ograničen. Dostupni su načini za tretiranje relacionih podataka, XML, JSON i drugih formata kao RDF, tako da se mogu vršiti SPARQL upiti nad podacima u ovim formatima – ili nad kombinacijom izvora, što je jedan od najmoćnijih aspekata SPARQL/RDF kombinacije [2].

Osnova SPARQL-a sastoji se iz upita u formi jednostavnih graf obrazaca. Takođe omogućava i niz funkcija za konstruisanje naprednih obrazaca upita, za navođenje dodatnih uslova za filtriranje i za formatiranje

prikaza konačnog rezultata. Ključne reči koje se najčešće javljaju u SPARQL upitu su: *PREFIX* (definiše namespace), *SELECT* (definiše format prikazanog rezultata) i *WHERE* (predstavlja uslove koji moraju biti zadovoljeni).

### 2.3.1. GeoSPARQL

Geoprostorni podaci su sve više dostupni na *web-u* u obliku skupova podataka opisanih uz pomoć RDF-a. Iako je SPARQL pogodan za upite nad vezama koje su eksplicitno predstavljene u podacima, nad implicitnim vezama, kao što su geoprostorne, ne mogu se tako lako vršiti upiti. Na primer, mogu postojati skupovi podataka koji opisuju spomenike i parkove, ali nije lako izvodljivo spojiti ta dva skupa na osnovu njihove nedefinisanе povezanosti [3].

Sposobnost davanja odgovora na smislen upit, kao što je “Koji parkovi su manje od 3 kilometra udaljeni od spomenika Svetozaru Miletiću u Novom Sadu?”, zavisi od toga kako su podaci predstavljeni, da li su svi resursi povezani sa spomenikom Svetozaru Miletiću, kao i da li je ta veza eksplicitno navedena.

*GeoSPARQL* je OGC (*Open Geospatial Consortium*) standard za predstavljanje i vršenje upita nad geoprostornim podacima u okviru semantičkog *web-a*. Definiše rečnik za predstavu geoprostornih podataka u RDF-u, kao i proširenje SPARQL upitnog jezika za njihovu obradu [4].

## 3. TEHNOLOGIJE

### 3.6. RDF4J

*Eclipse RDF4J* [5] predstavlja *open source* Java *framework* namenjen radu sa RDF podacima. Obuhvata parsiranje, skladištenje, rasuđivanje i izvršavanje upita nad RDF podacima. Obezbeđuje API koji se može povezati sa svim vodećim RDF skladištima. Dozvoljava povezivanje sa SPARQL *endpoint*-ima i kreiranje aplikacija koje se oslanjaju na *linked data* i semantički *web*.

RDF4J u potpunosti podržava SPARQL 1.1 upitni jezik za pisanje upita i nudi transparentni pristup udaljenim RDF repozitorijumima uz upotrebu istog API-ja kao za lokalni pristup. Takođe podržava najpoznatije formate RDF fajlova, uključujući RDF/XML, Turtle, N-Triples, N-Quads, JSON-LD, TriG i TriX.

Ono što je najvažnije pomenuti jeste da RDF4J nudi proširenu algebru za delimičnu podršku *GeoSPARQL-a*. Ovo podrazumeva dodatne geoprostorne funkcionalnosti kao deo SPARQL *engine-a*, na bilo kom RDF4J repozitorijumu, uz upotrebu *Spatial4J* i *JTS* biblioteka za geoprostorno zaključivanje. RDF4J podržava *GeoSPARQL* funkcije samo nad geoprostornim podacima predstavljenim u obliku WKT *string*-ova.

### 3.7. LeafletJS

*LeafletJS* je *open source JavaScript* biblioteka namenjena za implementaciju interaktivnih mapa u aplikacijama. Inicijalno je razvijena 2011. godine i podržava većinu mobilnih i desktop platformi. Izuzetno je kompaktna, svega 39KB JS koda, i sadrži funkcionalnosti mapiranja koje su od interesa većini developera.

Dizajniran je da podrži tri osnovna zahteva: jednostavnost, performanse i upotrebljivost. Može biti proširen sa velikim brojem *plugin-a*, a takođe je i detaljno pokriven dokumentacijom.

Podržava iscrtavanje tačaka, linija, poligona i drugih geometrijskih oblika, zbog čega je pogodan za potrebe ovog rada, a to su iscrtavanje i prikazivanje geoprostornih podataka učitanih preko *GeoSPARQL* upita.

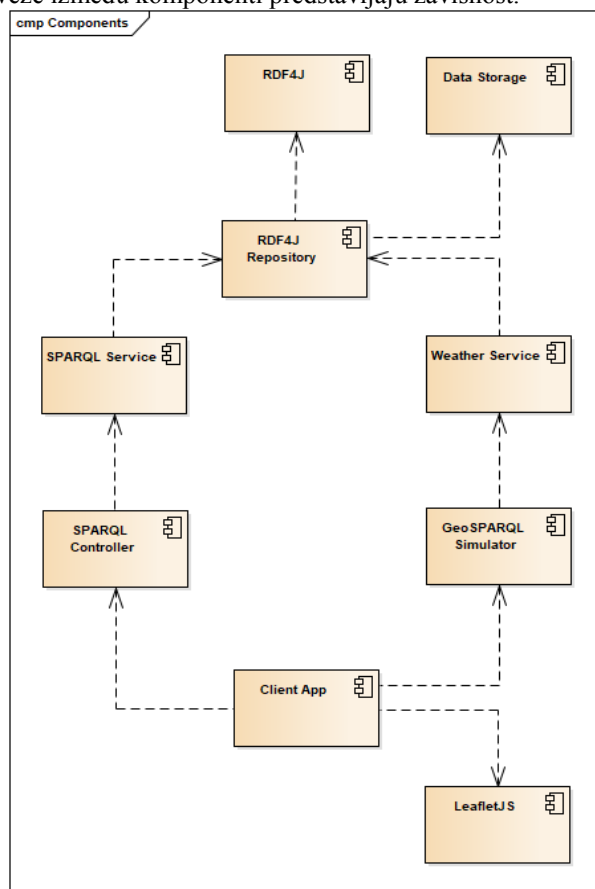
## 4. DIZAJN REŠENJA

### 4.1. Opis postojećeg rešenja – SPARQL Playground

*SPARQL Playground* aplikacija je osnova ovog rada na koju se nadograđuje. Predstavlja višenamensku *standalone web* aplikaciju namenjenu za učenje, predavanje i testiranje SPARQL upitnog jezika. Implementirana je u *Java Spring Boot* i *AngularJS*. Razvijena je na *SIB Swiss Institute of Bioinformatics*. Koristi *OpenRDF Sesame 2.8.6 framework* kao SPARQL *engine*.

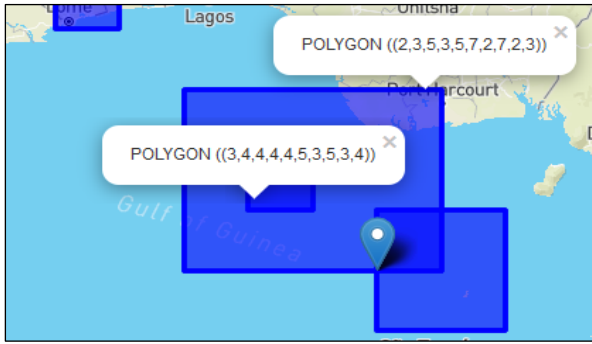
### 4.2. Dizajn proširenog rešenja

Proširena *SPARQL Playground* aplikacija se oslanja na RDF4J biblioteku kao SPARQL *engine* i koristi *LeafletJS* biblioteku za prikaz geoprostornih podataka. Dodatno, integriše se sa servisom vremenskih prilika [6] na osnovu čega je implementirana simulacija kretanja vremenskih nepogoda na mapi. Dijagram komponenti koji oslikava arhitekturu aplikacije je prikazan na *Slika 1*. Usmerene veze između komponenti predstavljaju zavisnost.



Slika 1: Dijagram komponenti proširene aplikacije

Na Slika 2 je prikazan primer poligona iscrtanih na osnovu WKT literala dobijenih iz rezultata SPARQL upita:



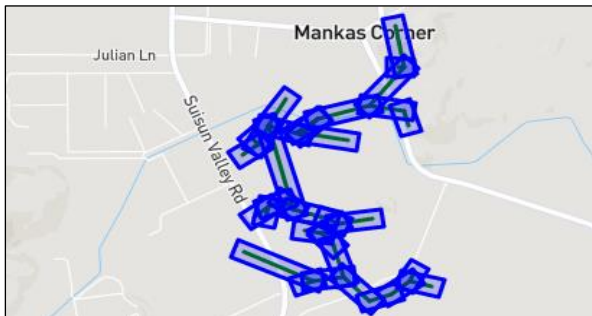
Slika 2: Primer iscrtanih poligona

Korisnik pored standardnih SPARQL upita, može da vrši *GeoSPARQL* upite nad geoprostornim podacima u obliku WKT literala. *OpenRDF Sesame 2.8.6 framework*, na koji se inicijalno oslanjao *SPARQL Playground*, ne podržava *GeoSPARQL* upite. Iz tog razloga je bilo neophodno migrirati aplikaciju da se oslanja na *RDF4J framework*, koji ga nasleđuje i proširuje *GeoSPARQL* podrškom.

#### 4.2.1. Grafički prikaz izvoda

Podržano je iscrtavanje tri vrste geometrijskih oblika: tačka, linija i poligona. Učitavanje CIMXML datoteka, gde svaka sadrži koordinate svih *ACLLineSegment*-a koji pripadaju tom izvodu, omogućava da se taj izvod grafički prikaže na mapi. Svaki *ACLLineSegment* sadrži referencu na svoj *Location*, a *Location* sadrži jedan ili više *PositionPoint*-a koji mu pripadaju.

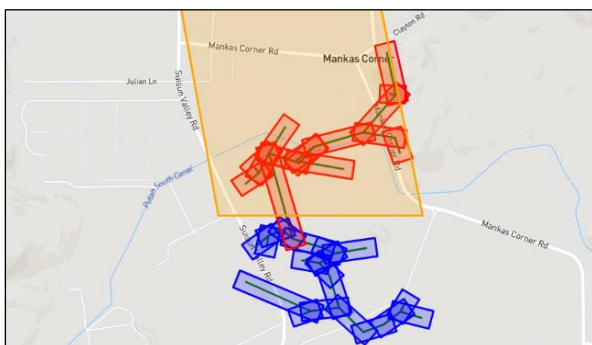
Slika 3 prikazuje primer iscrtanog izvoda zajedno sa poligonima koji predstavljaju njegovu zonu osetljivosti upotrebijenu prilikom rada servisa vremenskih prilika [6]:



Slika 3: Primer iscrtanog izvoda sa svojom zonom osetljivosti

#### 4.2.2. Servis vremenskih prilika

Primer rada servisa vremenskih prilika (*Weather service*) [6] prikazan je na Slika 4:



Slika 4: Simulacija kretanja oluje preko zone osetljivosti

Predstavlja servis u sklopu proširenja *SPARQL Playground*-a namenjen za simulaciju kretanja vremenskih nepogoda na mapi. Delovi terena koji su pogodile vremenske nepogode su predstavljeni kao poligoni. Ukoliko prilikom njihovog kretanja dođe do preklapanja sa nekom od zona osetljivosti izvoda, ta zona menja boju kao znak upozorenja.

## 5. IMPLEMENTACIJA

### 5.1. Migracija aplikacije na *RDF4J framework*

*OpenRDF Sesame 2.8.6 framework*, na koji se inicijalno oslanjao *SPARQL Playground*, ne podržava *GeoSPARQL*. *Eclipse RDF4J* ga nasleđuje i proširuje sa algebrom za delimičnu podršku *GeoSPARQL* upita. Ovo je bio razlog migracije aplikacije. Proširenje podrazumeva dodatne geoprostorne funkcionalnosti kao deo *SPARQL engine*-a, na bilo kom *RDF4J* repozitorijumu, uz upotrebu *Spatial4J* i *JTS* biblioteka za geoprostorno zaključivanje. *RDF4J* podržava *GeoSPARQL* funkcije samo nad geoprostornim podacima izraženim kao WKT literali.

Prvi korak migracije bio je uklanjanje *OpenRDF Sesame* zavisnosti (dependency) i njihova zamena novim *RDF4J* unutar *pom.xml* datoteke. Ceo *Sesame core framework* je bio uključen kao *sesame-runtime*. *Sesame-sail-native-rdf* zavisnost predstavlja *SAIL* implementaciju koja čuva podatke direktno kao datoteke na disku. *SAIL (Storage And Inference Layer)* je interfejs za skladištenje *RDF* podataka, preko kojih može da izvršava upite. Poslednja uklonjena zavisnost je *sesame-runtime-osgi*, koja obuhvata *OSGi (Open Service Gateway Initiative) runtime* zavisnosti za *OpenRDF* aplikaciju.

Nakon toga, zamenjeni su odgovarajućim *RDF4J* zavisnostima. Pre svega je dodat *BOM (Bill Of Materials)* koji služi da otkloni problem nepodudaranja verzija raznih uključenih zavisnosti. Aplikacija je zasnovana na 3.6.2 verziji *RDF4J*. Zavisnost *rdf4j-storage* obuhvata sve *RDF4J* biblioteke vezane za skladištenje i upite/zaključivanje. *Jackson API* je uključen kroz *jackson-core* (2.12.2) zavisnost i neophodan je za parsiranje i rad sa *JSON* podacima. Poslednja izmena u *pom.xml* datoteci je dodavanje *SLF4J API*-ja za logovanje u aplikaciji.

### 5.2. Grafički prikaz geoprostornih podataka

*LeafletJS* mapa se inicijalizuje prilikom pokretanja aplikacije. Geoprostorni podaci se prikazuju nakon izvršavanja *GeoSPARQL* upita ili nakon prelaska na stranicu *GeoSPARQL* Simulatora, gde se automatski učitavaju i prikazuju svi izvodi.

Podržano je prepoznavanje i obrada 4 tipa WKT literala:

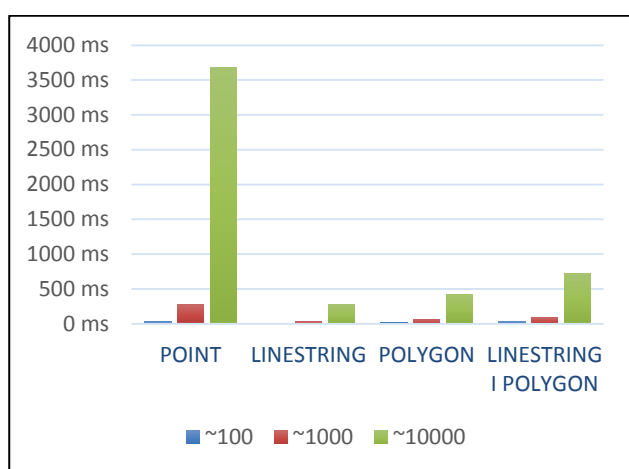
1. **POINT EMPTY** – Prazan element, nema grafičku reprezentaciju.
2. **POINT** – Predstavlja se kao jedna tačka na mapi.
3. **LINestring** – Predstavljen kao linija sa početnom i krajnjom tačkom. Izvodi se prikazuju uz pomoć linija, gde svaka od njih predstavlja jedan njegov *ACLLineSegment*.
4. **POLYGON** – Predstavljen kao niz tačaka gde su prva i poslednja jednake. Zone osetljivosti oko izvoda su prikazane poligonima, kao i vremenske nepogode u sklopu servisa vremenskih prilika.



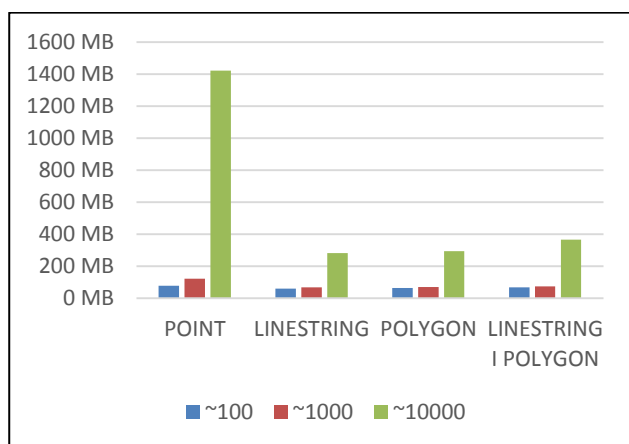
## 6. IZMEREENE PERFORMANSE

Performanse su merene na računaru sa sledećim karakteristikama: procesor *Intel Core i5-4460 3.20GHz*, 16GB RAM memorije, HDD sa kapacitetom od 1TB i *Microsoft Windows 10* operativni sistem. Testni podaci korišćeni prilikom merenja su bazirani na "*IEEE 37 Node Test Feeder*" [7] predstavljenim u CIMXML obliku.

Grafikon 1 prikazuje rezultate merenja performansi aplikacije prilikom iscrtavanja elemenata na mapi, a Grafikon 2 zauzeće memorije. Posmatrano je četiri slučaja: Iscrtavanje tačaka, linija, poligona i linija/poligona zajedno. Poslednji slučaj oslikava rad aplikacije prilikom iscrtavanja izvoda sa njihovim zonama osetljivosti. Svakom od ovih slučajeva je izmereno vreme iscrtavanja i zauzeće memorije 100, 1000 i 10000 elemenata, s tim što je broj elemenata dvostruko veći u slučaju linija i poligona zajedno.



Grafikon 1: Poređenje performansi iscrtavanja elemenata



Grafikon 2: Poređenje zauzeća memorije

Uočava se da su tačke daleko najslabije i po pitanju vremena iscrtavanja i po zauzeću memorije. Razlog ovome je što *LeafletJS* biblioteka tačke predstavlja *marker* elementima, koji su memorijski daleko zahtevniji od linija i poligona. Linije pokazuju najbolje performanse, a poligoni nešto slabije od njih.

## 7. ZAKLJUČAK

Migracija *SPARQL Playground*-a na *RDF4J framework* znatno proširuje potencijal ove aplikacije. Otvorena je

moćnost rada sa geoprostornih podacima uz pomoć *GeoSPARQL* upita, preko kojih se mogu izvesti zaključci o relacijama između geometrijskih entiteta. Integracija *LeafletJS* biblioteke dodatno proširuje aplikaciju podrškom prikaza geoprostornih podataka uz pomoć prostih elemenata na mapi.

Aplikacija se generalno pokazala dobro u slučajevima iscrtavanja do 10000 elemenata na mapi istovremeno, osim u slučaju prikaza 10000 tačaka. Potencijalno rešenje ovog problema bilo bi implementacija drugog načina za predstavljanje tačaka. Primer bi bio poligon ili kružnica manjih dimenzija. Drugo potencijalno proširenje može biti omogućavanje iscrtavanja kompleksnijih geometrijskih entiteta u WKT literal obliku. Treće i najveće proširenje može biti podrška selekcije 2 elementa na mapi, nakon čega se prikazuje meni sa skraćenicama za *GeoSPARQL* upite. Na taj način bi se za svaka dva selektovana elementa mogle dobiti informacije o njihovim međusobnim relacijama.

## 8. LITERATURA

- [1] JoshData, „<https://github.com>,” 2014. [Na mreži]. Available: <https://github.com/JoshData/rdfabout/blob/gh-pages/intro-to-rdf.md>. [Poslednji pristup 9 Jun 2021].
- [2] B. DuCharme, *Learning SPARQL*, drugo ur., O'Reilly Media, 2013.
- [3] R. Battle i D. Kolas, *GeoSPARQL: Enabling a Geospatial Semantic Web*, Knowledge Engineering Group, Raytheon BBN Technologies.
- [4] OGC *GeoSPARQL - A Geographic Query Language for RDF Data*, Open Geospatial Consortium, 2012.
- [5] „<https://rdf4j.org/>,” [Na mreži]. Available: <https://rdf4j.org/about/>. [Poslednji pristup 14 Jun 2021].
- [6] S. Stojaković, Testno okruženje za rukovanje meteorološkim podacima pomoću SPARQL upita, Novi Sad, 2021.
- [7] Distribution System Analysis Subcommittee, *IEEE 37 Node Test Feeder*, IEEE Power Engineering Society.

### Kratka biografija:



**Predrag Kaljević** rođen je 1994. godine u Novom Sadu. Završio je osnovnu školu „Jovan Jovanović Zmaj“ u Sremskoj Kamenici, 2009. godine. Završio je srednju ekonomsku školu „Svetozar Miletić“ u Novom Sadu, 2013. godine. Iste godine, u septembru, upisao je osnovne akademske studije na Fakultetu tehničkih nauka, smer softversko inženjerstvo i informacione tehnologije. Završio je osnovne studije 2019. godine i upisao master studije iste godine, smer primenjeno softversko inženjerstvo. Ispunio je sve obaveze i položio sve ispite predviđene studijskim programom.