

# Integração entre Sistemas Multiagente e a Plataforma Google Earth Engine para análise do fluxo d'água na Bacia Hidrográfica da Lagoa Mirim e Canal São Gonçalo

Fernanda P. Mota<sup>1</sup>, Marilton S. Aguiar<sup>2</sup>, Diana F. Adamatti<sup>3</sup>

<sup>1</sup>Universidade Católica de Pelotas (UCPEL)  
Pelotas – RS – Brazil

<sup>2</sup>Universidade Federal de Pelotas (UFPel)  
Pelotas – RS – Brazil

<sup>3</sup>Universidade Federal do Rio Grande (FURG)  
Rio Grande – RS – Brazil

{nandapm2010,dianaada}@gmail.com, marilton@inf.ufpel.edu.br

**Abstract.** *The main objective of this work is to use Multiagent Based Simulation (MABS) in conjunction with the Google Earth Engine platform to analyze the flow of rivers between regions. The simulation will be based on data from the state of Rio Grande do Sul, and focusing on the pilot application of the work in the Hydrographic Basin Management Committee. The Basin involves Lagoa Mirim and Canal São Gonçalo, specifically in the cities of Rio Grande and Pelotas. To the best of our knowledge, this methodology has not yet been applied in the context of the state, seeking a more interactive and participatory way of making decisions on water issues.*

**Resumo.** *Este trabalho tem como objetivo principal utilizar Simulação Baseada em Multiagente (MABS) em conjunto com a plataforma Google Earth Engine para analisar o fluxo dos rios entre as regiões. A simulação será feita a partir dos dados do estado do Rio Grande do Sul, e focando a aplicação-piloto do trabalho no Comitê de Gerenciamento das Bacias Hidrográficas. A Bacia envolve a Lagoa Mirim e do Canal São Gonçalo, especificamente nas cidades de Rio Grande e Pelotas. Do que rege nosso conhecimento, ainda não foi aplicada essa metodologia no contexto do estado, buscando uma forma mais interativa e participativa para a tomada de decisão sobre questões hídricas.*

## 1. Introdução

Os recursos hídricos desempenham um papel fundamental no crescimento econômico e sustentável da sociedade [Deng et al. 2019]. Os corpos d'água como lagoas, rios, córregos e reservatórios são componentes essenciais dos recursos hídricos e fornecem uma série de serviços como abastecimento de água e regulação climática [Kumar and Mutanga 2018]. No entanto, modificações na composição e na disponibilidade da água podem resultar em alterações nos ciclos de água [Wood et al. 2011]. Por isso, o monitoramento é necessário, pois qualquer mudança pode alterar a composição e os processos funcionais do ecossistema e a quantidade, qualidade e disponibilidade

d'água [Cann et al. 2013]. Devido a isso, podemos observar a importância de mapear a superfície da água para melhorar o gerenciamento da água em larga escala.

Antes do surgimento do sensoriamento remoto, o mapeamento em campo do solo era o único método disponível, que é um método caro, demorado e os resultados nem sempre eram precisos. Por outro lado, o sensoriamento remoto oferece um método espacial e temporal que permite o rastreamento de um registro histórico, que é benéfico para o mapeamento das informações da água em áreas que não possuem um monitoramento em terra. Na literatura podemos observar que houveram grandes esforços para monitorar a dinâmica da superfície da água por sensoriamento remoto, como: *i)* [McCullough et al. 2013] analisaram as imagens do satélite MODIS com resolução de 250 metros para monitoramento remoto de lagos no Maine nos Estados Unidos no período de 2000 a 2011; *ii)* [Feyisa et al. 2014] desenvolveram o AWEI (acrônimo de *Automated Water Extraction Index*) para melhoria da precisão da extração de água na presença de vários tipos de ruído ambiental; e, *iii)* [Fisher et al. 2016] propuseram métodos de índice de água baseados em dados normalizados para refletância da superfície, usando limiares otimizados para classificação automatizada de água em grandes regiões.

Na tentativa de auxiliar no processo de análise dos dados de sensoriamento remoto, a Google desenvolveu em 2005 a plataforma baseada em nuvem chamada *Google Earth Engine* (GEE)<sup>1</sup> que visa facilitar a análise em larga escala de dados geoespaciais [Mutanga and Kumar 2019]. O GEE está sendo largamente utilizado para análise de aplicações em larga escala, incluindo mapeamento de terrenos urbanos [Liu et al. 2018], arroz em casca [Dong et al. 2016], zonas úmidas [Hird et al. 2017], corpos d'água de superfície aberta [Pekel et al. 2016, Zou et al. 2018], entre outros. [Pekel et al. 2016] analisaram o mapeamento de massas de água de superfície aberta no mundo inteiro no período de 1984 a 2015, a partir da plataforma GEE. Os principais desafios computacionais no contexto da gestão de recursos naturais citados por [Fuller et al. 2007] seguem até hoje sendo problemáticas em aberto, são elas: *i)* gerenciamento e comunicação de dados, *ii)* análise de dados e *iii)* controle e otimização. Algumas soluções estão sendo propostas na área de Computação, a partir de técnicas da Inteligência Artificial como Sistemas Multiagente (SMA), Redes Neurais, Algoritmos Genéticos, Autômatos Celulares ou Inteligência de Enxames. Os modelos de simulação baseados em Agentes se sobressaem neste domínio pois apresentam soluções adequadas para cenários complexos, como: sistemas sociais, ecológicos e políticas públicas [Filatova et al. 2013]. Neste trabalho será utilizado o SMA para o desafio de simular um sistema ecológico relacionado ao fluxo de água dos rios.

Na literatura, observa-se um significativo número de ferramentas utilizadas em simulações baseadas em agentes, dentre elas podemos destacar: *i)* JaCaMo [Boissier et al. 2013] que é composto pelas ferramentas Jason [Bordini et al. 2007] e MOISE+ [Hübner et al. 2007]; *ii)* NetLogo que foi desenvolvido por [Tisue and Wilensky 1999]; *iii)* Cartago que foi desenvolvido por [Ricci et al. 2006] *iv)* CORMAS desenvolvida por [Page et al. 2000], e *v)* GAMA que foi desenvolvida por [Taillandier and Buard 2009] é utilizada para na análise de uso dos recursos hídricos. No entanto, nenhuma delas apresenta integração com a plataforma GEE. Sendo assim, neste trabalho propomos a integração de uma plataforma

---

<sup>1</sup><https://code.earthengine.google.com/>

de simulação sistema multiagente com a plataforma *Google Earth Engine*. Além disso, propomos um estudo de caso aplicado na plataforma GEE, a partir da base de dados da Secretaria Estadual de Meio Ambiente do estado do Rio Grande do Sul, no âmbito do Comitê de Gerenciamento das Bacias Hidrográficas da Lagoa Mirim e do Canal São Gonçalo, que envolvem as cidades de Rio Grande e Pelotas/RS.

O trabalho encontra-se organizado da seguinte forma: na Seção 2 são apresentados conceitos básicos sobre Sistemas Multiagente e *Google Earth Engine*; na Seção 3 apresentamos o sistema multiagente na plataforma GEE; por fim, a Seção 4 apresenta as conclusões e trabalhos futuros deste estudo.

## 2. Fundamentação Teórica

Nesta seção é apresentada a fundamentação teórica relacionada as principais temáticas envolvidas neste trabalho: *Google Earth Engine* e Sistemas Multiagente.

### 2.1. *Google Earth Engine*

O *Google Earth Engine* (GEE) foi desenvolvido pela Google em 2005 para processamento geoespacial baseado em nuvem. O GEE é uma plataforma que permite o acesso e a análise de forma eficiente de petabytes de dados geoespaciais a partir de imagens de sensoriamento remoto como *Landsat* e *MODIS*, entre outros. Estes dados podem ser acessados em servidores do *Google* em um período de tempo menor e a partir de computadores que não têm processamento [Gorelick et al. 2017]. O GEE utiliza um sistema de processamento paralelo para realizar o cálculo em um grande número de computadores. Este processamento beneficia-se de técnicas padrões que são frequentemente utilizadas em linguagens de programação funcionais como: transparência referencial e avaliação lenta, para ganhos significativos de otimização e eficiência [Gorelick et al. 2017].

O GEE possui um *front-end* de fácil acesso que proporciona um ambiente adequado para o desenvolvimento interativo de algoritmos e análise de dados. A plataforma possibilita a inserção e manipulação de código, de coleções e de dados do usuário. O processamento desta informação é feito por meio de recursos em nuvem do *Google*. O *Google Earth Engine* proporciona aos pesquisadores e entusiastas o acesso a um vasto banco de dados para que estes possam detectar mudanças climáticas ou na atmosfera, bem como mapeamento de tendências e quantificação de recursos na superfície da Terra [Kumar and Mutanga 2019]. O *Google* disponibilizou em 2008 o acesso gratuito a série de dados de satélite chamada *Landsat*. Além disso, a empresa disponibilizou gratuitamente o mecanismo de computação em nuvem para arquivar todos os conjuntos de dados de sensoriamento remoto para uso em código aberto. O arquivo também possui um conjunto de dados vetoriais baseados em sistemas de informações geográficas (*GIS – Geographic Information System*), modelos sociais, demográficos, climáticos, de elevação digital, camadas de dados climáticos, dados de outros satélites, entre outros [Gorelick et al. 2017].

O *Google Earth Engine* possibilita a interação do usuário com plataformas de diferentes formas. O GEE possibilita a construção de jogos e de aplicações *mobile* integrada com métodos de aprendizagem de máquina nas linguagens de programação *Python* e *Javascript*. O GEE é composto pelos seguintes módulos [Gorelick et al. 2017]: i) *Code Editor* é uma IDE para escrever e executar *scripts*; ii) *Explorer* é um aplicativo para explorar o catálogo de dados do GEE e executar análises simples; e, iii) bibliotecas cliente

em *Python* e *JavaScript* para auxiliar os usuários no desenvolvimento de seus protótipos. Em nossa revisão literária, realizada no *Google Scholar*, buscamos por estudos que relacionem o uso da plataforma GEE em conjunto com sistemas multiagente para análise e manipulação de dados relacionados aos recursos hídricos. No entanto, não foram encontrados trabalhos que envolvessem sistemas multiagente.

Na pesquisa apresentada em [Ou et al. 2020], geraram-se mapas de estufa multi-temporais a partir de imagens Landsat da região de preservação natural de Shouguang, China no período de 1990 a 2018 no GEE. Além disso, os autores avaliaram e quantificaram a dinâmica espaço-temporal e o aumento da agricultura nesta área de estudo. Os resultados demonstraram a vantagem de utilizar imagens do Landsat no GEE para monitorar o desenvolvimento de estufas em um período de longo prazo e forneceram uma perspectiva mais intuitiva para entender o processo dessa abordagem especial na produção agrícola. [Xia et al. 2019] investigaram as mudanças climáticas nas águas superficiais da bacia do rio Huai que apresentaram impactos significativos nas plantas agrícolas, no equilíbrio ecológico e no desenvolvimento socioeconômico. Para entender as mudanças nesta região, os autores utilizaram os dados disponíveis no Landsat TM, ETM+ e no OLI no período de 1989 a 2017 e processaram os dados na plataforma GEE. Segundo os autores, compreender as variações nas áreas de corpos d'água e os fatores de controle podem apoiar a designação e implementação de práticas sustentáveis no gerenciamento de água nos usos agrícola, industrial e doméstico.

No trabalho de [Shami and Ghorbani 2019] investigou-se o armazenamento de água superficial e subterrânea na região do Irã que tem clima quente e árido. Os autores analisaram o armazenamento de água a partir dos dados do satélite GRACE (do inglês, *Gravity Recovery and Climate Experiment*) e a precipitação anual total por meio do CHIRPS (do inglês, *Climate Hazards Group InfraRed Precipitation with Station Data*) no GEE no período de 2003 a 2017. Os resultados obtidos indicam uma redução nos níveis de armazenamento de água no período entre 2008 e 2017. Além disso, os dados de precipitação anual mostram que a quantidade de chuva diminuiu na região desde o ano de 2008.

## 2.2. Sistemas Multiagente

O modelo baseado em agentes consiste nas interações entre os agentes e o ambiente. Na simulação baseada em agentes (do inglês, *Agent-Based Simulation – ABS*) os agentes podem interagir uns com os outros, com pouca ou nenhuma coordenação centralizada [Azevedo and Meneze 2007]. Além disso, a ABS possibilita a interação dos agentes (pessoas, bactérias, insetos, nações ou organizações) entre si e com seu ambiente, o que possibilita a compreensão do comportamento que emerge a partir da comunicação entre eles. [Azevedo and Meneze 2007] destaca a importância do ABS para compreensão de sistemas complexos. [Russell and Norvig 2013] define agentes como entidades computacionais que são inseridas em um ambiente e que podem perceber (sensores) e atuar (atuadores) sobre o mesmo. Segundo o autor, um agente possui atributos para se controlar de forma autônoma, perceber o ambiente, persistir por um período, adaptar-se a mudanças e assumir metas [Russell and Norvig 2013].

Segundo [Azevedo and Meneze 2007], os agentes permitem a modelagem de situações nas quais os indivíduos têm comportamentos complexos e distintos, podendo

levar em conta tanto as propriedades quantitativas (parâmetros numéricos) quanto qualitativas (comportamento individual) do sistema. O autor ressalta que em um modelo simples que utiliza agentes é possível perceber o comportamento complexo que emerge de suas interações e que são importantes para a dinâmica do sistema. A ABS é utilizada para identificar as propriedades do modelo e com isso ganhar entendimento de um processo dinâmico, o que seria muito difícil de modelar com técnicas matemáticas padrões [Axelrod 1996]. É relevante destacar que os agentes cognitivos possuam três características: *cooperar, aprender e agir de maneira autônoma* de acordo com [Nwana 1996]. [Bordini et al. 2001] acrescentam ainda coordenação, competição e negociação como aspectos relevantes na concepção destes agentes. [Russell and Norvig 2013] define um sistema multiagente (SMA) como um conjunto de agentes que interagem entre si e que podem ser organizados em grupos.

A arquitetura amplamente utilizada no desenvolvimento de SMA é a BDI (do inglês, *Beliefs, Desires, Intentions*), baseada em um modelo cognitivo que representam, respectivamente, crenças, desejos e intenções [Rao et al. 1995]. Para [Wooldridge 2002], *crenças* é o que o agente percebe sobre si mesmo, os demais agentes e o ambiente no qual está inserido; *desejos* são os estados que o agente almeja atingir, geralmente são objetivos; e, *intenções* é a sequência de ações que determinado agente executa para alcançar um objetivo.

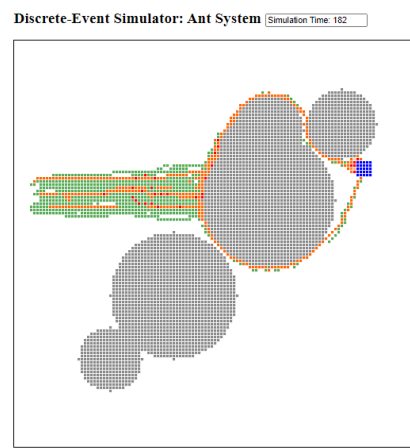
A maior diferença entre agentes cognitivos e reativos está relacionada a eficiência e a complexidade exigida do modelo de raciocínio do agente, ou seja, se o modelo de raciocínio exigir pouca complexidade e grande eficiência pode-se utilizar agentes reativos, pois necessitam de pouca memória e complexidade. Por outro lado, se o modelo necessitar de raciocínio e comportamentos complexos, a melhor alternativa seria o uso de agentes cognitivos, que apresentam memória e podem planejar suas ações [Dimuro et al. 2005]. Contudo, as propriedades, a arquitetura e a estrutura dos agentes, bem como o ambiente em que estes estão inseridos, são implementados de acordo com o problema a ser resolvido, a complexidade e o domínio específico de cada aplicação. Em [Russell and Norvig 2013], diversas aplicações e algoritmos são apresentados de forma a exemplificar a busca de soluções para esta demanda crescente de problemas.

Ao retomarmos a ideia inicial de nossa pesquisa, que é a análise e auxílio no processo de gestão de recursos hídricos, as aplicações que utilizam Sistemas Multiagente para modelagem e simulação do gerenciamento de recursos têm se mostrado uma possibilidade efetiva na busca por soluções para essa problemática. O SMA apresenta os seguintes benefícios que podem auxiliar nesta questão: *i)* rapidez na resolução de problemas devido ao processamento concorrente; *ii)* aumento na flexibilidade e na escalabilidade através da conexão de vários sistemas; *iii)* aumento na capacidade de resposta a um determinado problema pelo fato de todos os recursos estarem localizados no mesmo ambiente [Alvares and Sichman 1997].

### 2.3. js-Simulator

O *js-Simulator* (Jssim) é uma biblioteca que foi construída em *javascript* a partir da *MASON - Multiagent Simulation* que é uma biblioteca em Java. O Jssim é um simulador multiagente de eventos discretos de uso geral para modelagem e simulação baseada em agentes [Chen 2017].

O Jssim é uma biblioteca de simulação de eventos discretos, de processo único, de uso geral, destinada a oferecer suporte a diversas experiências multiagentes, desde robôs contínuos 3D a redes de complexidade social, até algoritmos discretos de simulação de comportamento de enxames [Chen 2017]. Na Figura 1 podemos observar a simulação do comportamento de uma colônia de formigas ao procurar comida. Embora cada formiga siga um conjunto de regras simples, a colônia como um todo age de maneira sofisticada. Quando uma formiga encontra um pedaço de comida, ela a leva de volta ao formigueiro e exala ferormônio à medida que se move. Quando outras formigas sentem o ferormônio, elas o seguem em direção à comida. À medida que mais formigas transportam comida para o formigueiro, elas reforçam a trilha do ferormônio [Luke et al. 2003].



**Figura 1. Exemplo do comportamento de um formigueiro usando a biblioteca Jssim.**

O objetivo do Jssim é facilitar a criação de uma ampla variedade de ambientes de simulação com vários agentes. O Jssim é uma biblioteca que possibilita o desenvolvimento de ferramentas especializadas que podem ser utilizadas em diferentes tarefas. O Jssim fornece uma biblioteca de modelos do simulador, e uma ferramenta para visualizar e manipular o modelo por meio de uma interface gráfica [Luke et al. 2003].

O simulador e a biblioteca de modelos foram desenvolvidos separadamente, o que permite a implementação em *back-end* dos modelos enquanto o usuário pode visualizar e verificar suas simulações no experimento. O modelo pode ser serializado e recuperado do armazenamento a qualquer momento, e o sistema de visualização pode ser adicionado ou removido do modelo a qualquer momento. As execuções podem ser repetidas em qualquer plataforma com resultados idênticos [Luke et al. 2003].

A biblioteca de modelos Jssim contém uma programação de eventos discretos para representar o tempo, além de várias representações espaciais chamadas vizinhanças. O Jssim não possui um conjunto de modelos espaciais prescritos, atualmente a biblioteca vem com modelos simples em espaços 2D discretos, 2D hexagonais e 2D contínuos. Qualquer objeto pode ser armazenado nessas vizinhanças e os modelos podem ser utilizados em qualquer combinação e qualquer número em uma determinada simulação [Chen 2017].

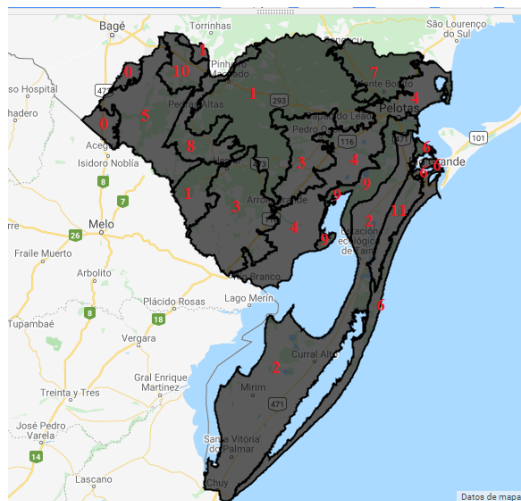
O Jssim separa a ideia de “agente” da corporificação, ou seja, agentes são simplesmente objetos que podem ser programados para serem executados. Quando executados, os agentes geralmente manipulam objetos armazenados em sua vizinhança. Como qual-

quer outro objeto, os agentes podem ser inserido em sua proximidade se isso for apropriado para a simulação. O Jssim contém vários exemplo embutidos em sua biblioteca, incluindo colônia de formigas, *Flocking Boids* em 2D contínuos, modelos contínuos de simulação de infecção por vírus, entre outros [Chen 2017].

### 3. Sistema Multiagente no Google Earth Engine

Neste trabalho foi desenvolvida a modelagem inicial para o fluxo d'água nas regiões próximas a bacia hidrográfica da Lagoa Mirim e Canal São Gonçalo por meio de sistemas multiagente. Na Figura 2 podemos observar a integração dos agentes reativos do modelo do SMA no GEE representado pelas doze regiões da bacia, as quais foram obtidas a partir dos dados georreferenciados cedidos pela Secretaria Estadual de Meio Ambiente do estado do Rio Grande do Sul, no âmbito do Comitê de Gerenciamento das Bacias Hidrográficas da Lagoa Mirim e do Canal São Gonçalo, que envolvem as cidades de Rio Grande e Pelotas/RS.

O modelo foi desenvolvido em *javascript* na plataforma GEE a partir da biblioteca Jssim, no qual os agentes são eventos que podem ser programados para serem executados em um determinado período. O simulador de eventos discretos é gerenciado no Jssim por meio da classe *Scheduler*, que programa e dispara eventos com base em sua especificação de tempo e classificação, ou seja, de acordo com ordem do evento. A lógica principal de um evento é definida em seu método de atualização (*deltaTime*), ou seja os eventos com classificação maior. Eventos com prioridade alta e tempo de disparo precedente serão executados primeiro pelo *Scheduler*. Um evento também pode ser planejado para disparar em um momento posterior a partir do horário atual. Os eventos com a mesma classificação serão embaralhados antes da execução.



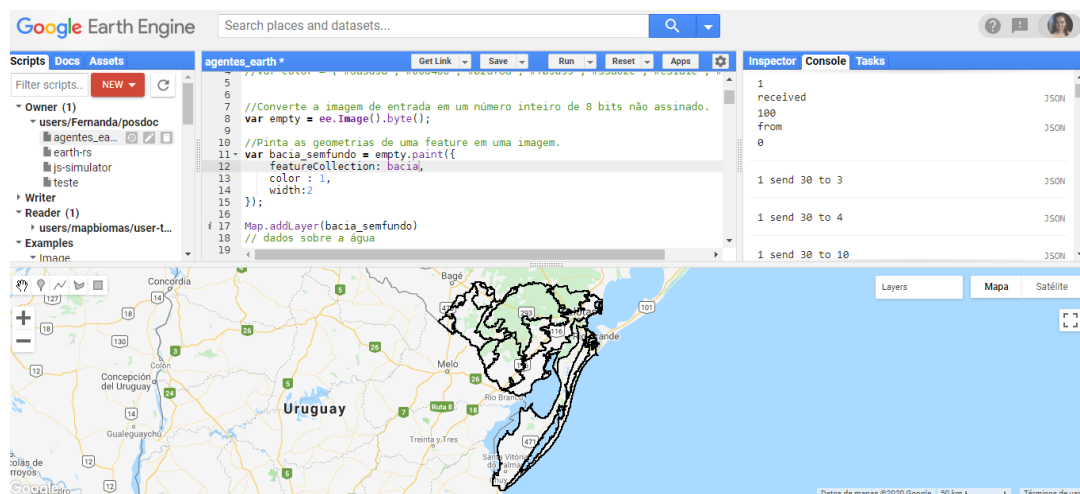
**Figura 2. Regiões ao redor da Bacia hidrográfica da Lagoa Mirim e Canal São Gonçalo.**

Na Figura 3 podemos observar o modelo multiagente implementado na plataforma GEE. Neste modelo, cada região representa um agente reativo, que pode enviar água para os vizinhos próximos, ou seja, não há uma tomada de decisão cognitiva. A Tabela 1 demonstra a direção que o fluxo d'água pode seguir de um agente em direção ao outro. As colunas representam os agentes e cada linha descreve com quais regiões cada agente

**Tabela 1. Matriz de fluxo d'água entre as regiões.**

	R1	R2	R3	R4	R5	R6	R7	R8	R9	R10	R11	R12
R1	0	0	0	0	0	0,9	0	0	0	0	0	0
R2	0	0	0	0,3	0,3	0	0	0	0	0	0,3	0
R3	0	0	0	0	0	0	0,4	0	0	0	0	0,5
R4	0	0	0	0	0,9	0	0	0	0	0	0	0
R5	0	0	0	0	0	0	0	0	0	0,4	0	0,5
R6	0	0,4	0	0	0	0	0	0	0,5	0	0	0
R7	0	0	0,4	0	0	0	0	0	0	0	0	0,5
R8	0,4	0	0	0	0,5	0	0	0	0	0	0	0
R9	0	0,4	0	0,5	0	0	0	0	0	0	0	0
R10	0	0	0,9	0	0	0	0	0	0	0	0	0
R11	0	0	0	0	0	0,4	0	0	0,5	0	0	0
R12	0	0	0	0	0	0	0,9	0	0	0	0	0

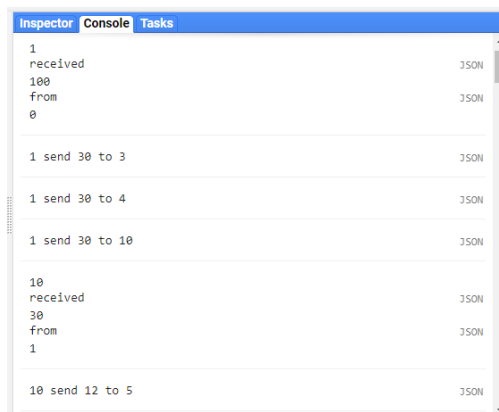
pode se comunicar. Os agentes enviam uma mensagem avisando a quantia de água que receberam e a quantia que estão enviando para os vizinhos. Neste trabalho consideramos que um agente envia 90% da água que recebe para os outros agentes e mantém 10% em sua região.

**Figura 3. Modelo multiagente implementado na plataforma GEE.**

Na Figura 4 podemos observar a troca de mensagens entre as regiões da Bacia hidrográfica da Lagoa Mirim e Canal São Gonçalo sobre a quantidade de água que receberam e quanto de água estão enviando e para qual região. O fluxo de água entre essas regiões foi desenvolvido a partir da Tabela 1.

Neste trabalho foi possível observar que a plataforma GEE propicia a integração de sistemas de dados georreferenciados, o que a torna interessante no estudo de caso da bacia hidrográfica da Lagoa Mirim e Canal São Gonçalo no âmbito da área de recursos hídricos. Além disso, podemos analisar o fluxo d'água entre agente que representam regiões da bacia, o que nos permite contribuir com a implementação de sistemas multiagente na plataforma GEE, uma vez que não encontramos trabalhos na literatura que integrem o uso do GEE com sistemas multiagente.





**Figura 4. Troca de mensagens entre as regiões da Bacia hidrográfica da Lagoa Mirim e Canal São Gonçalo.**

#### 4. Conclusão e Trabalhos Futuros

Os recursos e serviços hídricos desempenham um papel crucial no crescimento econômico e na sustentabilidade ambiental. Devido a isso, precisamos melhorar a coleta de dados hidrológicos, sua análise e o entendimento dos processos físicos da água. Esta pesquisa avaliou a utilização de Simulação Baseada em Multiagente (MABS – *Multiagent Based Simulation*) na gestão participativa dos recursos hídricos, mais especificamente, utilizando como base os dados do estado do Rio Grande do Sul e focando a aplicação-piloto do trabalho no Comitê de Gerenciamento das Bacias Hidrográficas da Lagoa Mirim e do Canal São Gonçalo, que envolvem as cidades de Rio Grande e Pelotas.

Este trabalho tem como objetivo principal apresentar as funcionalidades da plataforma Google Earth Engine (GEE), tendo como objetivos específicos identificar e avaliar como a plataforma pode auxiliar no contexto de análise de dados em recursos hídricos, e como realizar sua integração a um sistema multiagente. Nossa principal contribuição foi o desenvolvimento de Sistema Baseado em Multiagente a partir de dados georreferenciados no GEE para analisar o fluxo dos rios entre as regiões a partir dos dados do estado do Rio Grande do Sul.

Como resultado, podemos analisar o fluxo d'água entre as diferentes regiões a partir de um sistema multiagente que permite a troca de mensagens entre as regiões. Assim, acreditamos que este trabalho pode auxiliar na gestão participativa dos recursos hídricos ao fornecer informações da bacia e um modelo para que os gestores possam avaliar o fluxo d'água e possíveis problemas e conflitos que esses dados podem gerar ao longo das regiões da bacia.

Como trabalhos futuros propomos: *i*) realizar um teste de interface e de funcionalidade, de forma a avaliar quais funcionalidades devemos melhorar no visualizador e no simulador; *ii*) alterar as cores das regiões conforme o fluxo d'água vai passando de uma região para outra.

#### Agradecimentos

Os autores deste artigo agradecem a Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES/Brasil) e a Agência Nacional de Águas (ANA/Brasil) – Pró-

Recursos Hídricos Chamada N° 16/2017, pelo auxílio financeiro no desenvolvimento desta pesquisa.

## Referências

- Alvares, L. O. and Sichman, J. S. (1997). Introdução aos sistemas multiagentes. In *XVII Congresso da SBC-Anais JAI'97*.
- Axelrod, R. (1996). The complexity of cooperation: agent-based models of competition and collaboration. *Princeton Univ. Press*.
- Azevedo, L. L. and Meneze, C. S. (2007). Netplay – uma ferramenta para construção de modelos de simulação baseado em multiagente. *XVIII Simpósio Brasileiro de Informática na Educação, SBIE, Mackenzie*.
- Boissier, O., Bordini, R. H., Hübner, J. F., Ricci, A., and Santi, A. (2013). Multi-agent oriented programming with jacamo. *Science of Computer Programming*, 78(6):747–761.
- Bordini, R. H., Hübner, J. F., and Wooldridge, M. (2007). *Programming Multi-Agent Systems in AgentSpeak using Jason*. John Wiley & Sons.
- Bordini, R. H., Vieira, R., and Moreira, A. F. (2001). Fundamentos de sistemas multiagentes. In *Anais do XXI Congresso da Sociedade Brasileira de Computação (SBC2001)*, volume 2, pages 3–41.
- Cann, K., Thomas, D. R., Salmon, R., Wyn-Jones, A., and Kay, D. (2013). Extreme water-related weather events and waterborne disease. *Epidemiology & Infection*, 141(4):671–686.
- Chen, X. (2017). js-simulator. <https://github.com/chen0040/js-simulator>.
- Deng, Y., Jiang, W., Tang, Z., Ling, Z., and Wu, Z. (2019). Long-term changes of open-surface water bodies in the yangtze river basin based on the google earth engine cloud platform. *Remote Sensing*, 11(19):2213.
- Dimuro, G. P., Costa, A. C. R., and Palazzo, L. A. M. (2005). Systems of exchange values as tools for multi-agent organizations. *Journal of the Brazilian Computer Society*.
- Dong, J., Xiao, X., Menarguez, M. A., Zhang, G., Qin, Y., Thau, D., Biradar, C., and Moore III, B. (2016). Mapping paddy rice planting area in northeastern asia with landsat 8 images, phenology-based algorithm and google earth engine. *Remote sensing of environment*, 185:142–154.
- Feyisa, G. L., Meilby, H., Fensholt, R., and Proud, S. R. (2014). Automated water extraction index: A new technique for surface water mapping using landsat imagery. *Remote Sensing of Environment*, 140:23–35.
- Filatova, T., Verburg, P. H., Parker, D. C., and Stannard, C. A. (2013). Spatial agent-based models for socio-ecological systems: Challenges and prospects. *Environmental modelling & software*, 45:1–7.
- Fisher, A., Flood, N., and Danaher, T. (2016). Comparing landsat water index methods for automated water classification in eastern australia. *Remote Sensing of Environment*, 175:167–182.

- Fuller, M. M., Wang, D., Gross, L. J., and Berry, M. W. (2007). Computational science for natural resource management. *Computing in Science & Engineering*, 9(4):40.
- Gorelick, N., Hancher, M., Dixon, M., Ilyushchenko, S., Thau, D., and Moore, R. (2017). Google earth engine: Planetary-scale geospatial analysis for everyone. *Remote sensing of Environment*, 202:18–27.
- Hird, J. N., DeLancey, E. R., McDermid, G. J., and Kariyeva, J. (2017). Google earth engine, open-access satellite data, and machine learning in support of large-area probabilistic wetland mapping. *Remote sensing*, 9(12):1315.
- Hübner, J. F., Sichman, J. S., and Boissier, O. (2007). Developing organised multiagent systems using the MOISE+ model: programming issues at the system and agent levels. *Int. J. Agent-Oriented Software Engineering*, 1(3/4):370–395.
- Kumar, L. and Mutanga, O. (2018). Google earth engine applications since inception: Usage, trends, and potential. *Remote Sensing*, 10(10):1509.
- Kumar, L. and Mutanga, O. (2019). *Google Earth Engine Applications*. MDPI.
- Liu, X., Hu, G., Chen, Y., Li, X., Xu, X., Li, S., Pei, F., and Wang, S. (2018). High-resolution multi-temporal mapping of global urban land using landsat images based on the google earth engine platform. *Remote sensing of environment*, 209:227–239.
- Luke, S., Balan, G. C., Panait, L., Cioffi-Revilla, C., and Paus, S. (2003). Mason: A java multi-agent simulation library. In *Proceedings of Agent 2003 Conference on Challenges in Social Simulation*, volume 9.
- McCullough, I. M., Loftin, C. S., and Sader, S. A. (2013). Lakes without landsat? an alternative approach to remote lake monitoring with modis 250 m imagery. *Lake and reservoir management*, 29(2):89–98.
- Mutanga, O. and Kumar, L. (2019). Google earth engine applications.
- Nwana, H. S. (1996). Software agents: An overview. *The knowledge engineering review*, 11(3):205–244.
- Ou, C., Yang, J., Du, Z., Liu, Y., Feng, Q., and Zhu, D. (2020). Long-term mapping of a greenhouse in a typical protected agricultural region using landsat imagery and the google earth engine. *Remote Sensing*, 12(1):55.
- Page, C. L., Bousquet, F., Bakam, I., Bah, A., and Baron, C. (2000). CORMAS : A multiagent simulation toolkit to model natural and social dynamics at multiple scales. In *Wageningen : Resource Modeling Association*.
- Pekel, J.-F., Cottam, A., Gorelick, N., and Belward, A. S. (2016). High-resolution mapping of global surface water and its long-term changes. *Nature*, 540(7633):418–422.
- Rao, A. S., Georgeff, M. P., et al. (1995). Bdi agents: from theory to practice. In *Icmas*, volume 95, pages 312–319.
- Ricci, A., Viroli, M., and Omicini, A. (2006). Cartago: A framework for prototyping artifact-based environments in mas. In *International Workshop on Environments for Multi-Agent Systems*, pages 67–86. Springer.
- Russell, S. and Norvig, P. (2013). *Inteligência Artificial*. Elsevier Ltda, Rio de Janeiro/RJ, 3a edition.

- Shami, S. and Ghorbani, Z. (2019). Investigating water storage changes in iran using grace and chirps data in the google earth engine system. *International Archives of the Photogrammetry, Remote Sensing & Spatial Information Sciences*.
- Taillandier, P. and Buard, E. (2009). Designing agent behaviour in agent-based simulation through participatory method. In *International Conference on Principles and Practice of Multi-Agent Systems*, pages 571–578. Springer.
- Tisue, S. and Wilensky, U. (1999). Center for connected learning and computer-based modeling northwestern university, evanston, illinois. *NetLogo: A Simple Environment for Modeling Complexity*, Citeseer.
- Wood, E. F., Roundy, J. K., Troy, T. J., Van Beek, L., Bierkens, M. F., Blyth, E., de Roo, A., Döll, P., Ek, M., Famiglietti, J., et al. (2011). Hyperresolution global land surface modeling: Meeting a grand challenge for monitoring earth’s terrestrial water. *Water Resources Research*, 47(5).
- Wooldridge, M. (2002). An introduction to multi agent systems, department of computer science, university of liverpool, uk.
- Xia, H., Zhao, J., Qin, Y., Yang, J., Cui, Y., Song, H., Ma, L., Jin, N., and Meng, Q. (2019). Changes in water surface area during 1989–2017 in the huai river basin using landsat data and google earth engine. *Remote Sensing*, 11(15):1824.
- Zou, Z., Xiao, X., Dong, J., Qin, Y., Doughty, R. B., Menarguez, M. A., Zhang, G., and Wang, J. (2018). Divergent trends of open-surface water body area in the contiguous united states from 1984 to 2016. *Proceedings of the National Academy of Sciences*, 115(15):3810–3815.