Regional Program STIC-AmSud
2017 Project Proposal (Research – Innovation)

Basic Form

- This form, and the associated CVs, must be filled in English. Before filling the form, please read carefully the bases published in the STIC-AmSud site (www.sticamsud.org).
- This form must be sent in **.pdf** by email to the STIC-AmSud Secretariat (stic@sticamsud.org) by the project's International Coordinator.

A. General Information

A1	**Project title**
	Detection strategies based on Software Metrics for Multitier JavaScript

A2	**Acronym**
	DAJA

A3	**Research domain**
	Software engineering

A4	**Project goals**
	The DAJA project will extend the state-of-the-Art in software quality for the JavaScript ecosystem. In particular, DAJA will reach the following goals: • Provide meta-modeling and visualization techniques to analyze the static and dynamic structure of JavaScript applications. Our approach will be tunable and easily customizable to focus on a particular domain when necessary. • Identify and characterize recurrent situations in JavaScript source code leading to software code anomalies (*i.e.,* static or dynamic suboptimal structure) and vulnerable code. • Conduct relevant experiments over an adequate and representative subset of the JavaScript ecosystem, including analyzing software systems and conducting experiments involving developers (considering open and close source code). In particular, our approach will be available to Hop.js, an extension of JavaScript. • Provide software artifacts to make our results accessible to the software engineering community. The resulting tools will be released under an open source license.

A5	Abstract
	JavaScript is the most popular object scripting programming language. It is extensively used in web pages, servers, and web applications, among others. While JavaScript was initially conceived only for scripting, it is frequently used in large applications. The rapid adoption of JavaScript has outpaced the Software Engineering community to propose solutions to ensure a satisfactory code quality production. This situation has favored the production of poor quality JavaScript applications: we have found across JavaScript applications a large presence of dead-code (*i.e.,* source code portion that is never used) and code duplications. These symptoms are known to lead to maintenance and performance degradation. Moreover, we have previously analyzed potential security threats to JavaScript applications produced by bad coding practices [18]. The DAJA project will provide methodologies, techniques, and tools to ease the maintenance of software applications written in JavaScript while improving its security. DAJA is structured around two sequential phases: • In a first phase, we will provide an open JavaScript meta-model that exhibit structural properties and supports software metrics. Polymetric views are a family of software visualizations in which software metrics are mapped to a software structure. We will associate polymetric views to our meta-model as a foundation of DAJA. • In a second phase, we will build analyzing tools to identify different sources of code anomalies (*i.e.,* portion of code contributing to technical debt [8]) and vulnerable code (*i.e.,* portion of code that are potential security threats). Specifically, we will identify, measure, and characterize two attributes known to contribute to the software quality deterioration: dead code (*i.e.,* portion of code that is never exercised for a given set of program execution), and the top ten OWASP vulnerabilities[1] (a consensus of the most critical web application security flaws). We also plan to identify other types of anomalies that can affect the architecture of the applications. The produced analyzing tools will be evaluated on a representative corpus of JavaScript applications. Moreover, our analyzing tools will be designed to be extensible. In particular, our tools will operate with the Hop.js JavaScript engine. Hop.js is an extension of JavaScript in which a single code portion may execute on the client and the server. Applying our techniques to Hop.js will increase our visibility and exposes our analyzes to industrial partners. The research areas explored by DAJA are new within the context of JavaScript. The software artifacts produced by DAJA will be available under an open-source license. Three research teams will collaborate on DAJA: (i) the INDES project, based in INRIA Sophia-Antipolis, having an expertise in JavaScript engine and web application security; (ii) the ISIS-TAN institute, based in Argentina, having an expertise in identifying and prioritizing code anomalies; (iii) the Pleaid laboratory, based at the University of Chile, having an expertise in software visualization and JavaScript analyzing.

[1] `https://www.owasp.org/index.php/Category:OWASP_Top_Ten_Project`

A6	Scientific coordinators at each institution			
	South America A		**South America B**	
	Institution	University of Chile	Institution	UNICEN University
	Project coordinator	Dr. Alexandre Bergel	Project coordinator	Dr. Santiago Vidal
	Address	Beauchef 851, Of 308, Santiago, Chile	Address	Campus Universitario, Pje. Arroyo Seco (B7001BBO) Tandil, Buenos Aires, Argentina
	Phone/Fax	Phone: +56 9 913 123 81	Phone/Fax	Phone: +54 9 2494 507504
	Email	abergel@dcc.uchile.cl	Email	svidal@exa.unicen.edu.ar
	South America C		**South America D**	
	Institution		Institution	
	Project coordinator		Project coordinator	
	Address		Address	
	Phone/Fax		Phone/Fax	
	Email		Email	
	France A		**France B**	
	Institution	INDES/Inria	Institution	
	Project coordinator	Dr. Manuel Serrano	Project coordinator	
	Address	INRIA Sophia Antipolis, 2004 route des Lucioles - BP 93, F-06902 Sophia Antipolis, Cedex, France	Address	
	Phone/Fax	Phone: +33 4 92 38 76 39	Phone/Fax	
	Email	Manuel.Serrano@inria.fr	Email	

A7	Other participating institutions	
	In South America	**In France**

A8	List of expected participants (name and affiliation)
	Alexandre Bergel (general coordinator), researcher, University of Chile Alejandro Infante, Master Student, University of Chile Juraj Kubelka, PhD student, University of Chile Diego Orellana, University of Chile Santiago Vidal (local coordinator), researcher, UNICEN University Andrés Diaz Pace, researcher, UNICEN University Claudia Marcos, researcher, UNICEN University Hernan Ceferino Vazquez, PhD student, UNICEN University Manuel Serrano (local coordinator), researcher, INDES/Inria Tamara Rezk , researcher, INDES/Inria Colin Vidal, PhD student, INDES/Inria Francis Some, PhD student, INDES/Inria

A9	International Project Coordinator (to be chosen among the Scientific Coordinators mentioned in A6)
	Dr. Alexandre Bergel, University of Chile

B. Project Details

B1. Project guidelines

The DAJA project will provide techniques and tools to identify, measure, and characterize situations that threat the evolution and reduce the security of JavaScript applications. DAJA is structured around five work packages and their completion will be carried out over two years. DAJA involves three research groups located in Argentina, Chile, and France.

B2. Project description

JavaScript is the most popular client-side scripting language for Web applications [14]. Despite its great success, JavaScript developers have little tool support to identify and deal with sources of code anomalies and vulnerabilities. The need for better tools to help to understand, control and develop better JavaScript applications is highlighted in several works [11, 13, 15]. However, the automatic identification and analysis of patterns that threat the evolution and security of JavaScript applications have not been carefully investigated. The DAJA project wants to improve along this line by providing techniques to identify recurrent situations that lead to suboptimal static and dynamic structures.

In order to achieve the project goals, we will base our research work on an iterative development schema. We will alternate among adding/extending functionality of the existing tools, incorporating identification of anomalous structures and removal techniques, and exercising the tools on case-studies. This mode of work will help us to assess risks and benefits of the developed techniques, making the necessary adjustments. The research will be conceptually divided into 5 work packages, namely:

WP 1 – Defining a meta-model: Moose[2] [17] is a software analysis platform designed to provide the necessary infrastructure for formulating queries and computing software metrics. Moose supports multiple languages via the FAMIX meta-model [9]. FAMIX is a family of meta-models to describe program source code. While FAMIX have representations for different programming languages like Java, C++, and Smalltalk, no meta-model exist for JavaScript.

This work package will provide an extended representation of JavaScript applications. That is, the functions, variables, and modules that compose an application and the relationships between them. We will extend the FAMIX meta-model of Moose. The JavaScript meta-model will allow us to formulate software queries on the Moose platform about any JavaScript application. In this way, the meta-model will be the base for the following work packages.

WP 2 – Visualizing JavaScript applications using polymetric views: Polymetric view [16] have been proposed as a solution to visually map metrics and properties on a software structure. Polymetrics views have been applied to address a wide range of software maintenance issues, including visualizing component dependencies [5] and monitoring performance evolution [1].

In this work package, we will apply the "class complexity" visual metaphor [10] and turn it into a visual framework on which particular metrics and structural properties may be plugged-in. This extended class complexity will form the bases of the analyzing tools used in the following work packages. In particular, class complexity will be used as the main medium on which code duplication, and dead code will be represented. The visualization will be interactive to let practitioner easily identify, and navigate through the code portion culprit of the anomaly.

WP 3 – Identifying anomalous code: Code anomalies (also known as code smells) are a popular mechanism to identify signs of technical debt [12]. In this way, an anomaly acts as an anti-pattern indicating code that should be improved [19, 23]. Each anomaly affects one ore more components (*e.g.,* packages, classes, methods) of a system. Some of the symptoms used by code anomalies include: duplicated and unreachable code, very large methods or classes, long lists of parameters or violations in the encapsulation of a class, among others.

[2] `http://moosetechnology.org`

This work package will focus on the definition of patterns to identify code anomalies. Specifically, we will define patterns for dead (and unreachable) code, duplicated code, and redundant computation. The patterns will be identified statically using Moose based on the meta-model defined in WP 1 and dynamically by using the framework Aran [6]. The proposed techniques to identify the patterns will be implemented in an automatic tool.

WP 4 – Application to security and vulnerable code: The Open Web Application Security Project (OWASP)[3] is a well known organization focused on improving the security of software. OWASP regularly publishes the lists of "The Ten Most Critical Web Application Security Risks"[4]. The goal of this list to make critical industrial sectors (*e.g.,* financial, healthcare, defense, energy) aware of potential risks and threats. Most of these security risks are due to bad and improper coding style.

In this work package, we will apply the result of the previous work packages to identify vulnerabilities in software systems. We argue that detection strategies based on source code structure and metrics can be successively expressed to well-known source code vulnerabilities. By analyzing the source code of an application, these rules can be used to check, detect, and search for occurrences of different patterns that indicate a particular vulnerability. As a consequence, a vulnerability can be expressed as a rule combining different metrics, which have to reach predetermined thresholds. The thresholds values will be determined empirically (*e.g.,* based on statistical information obtained from practical experiences).

WP 5 – Application to Hop.js: Hop.js is a programming language that differentiates from other JavaScript frameworks and implementations by supporting multitier programming. In the multitier programming paradigm a single language is used to describe the server-side and client-side components of the applications. This enables designing of global analyses and tools that report on the global executions, by opposition to analyses that only cope with client-side executions. Combining the results of the other work packages and this unique Hop.js feature we will develop global JavaScript analyses, that will not only improve the performances but also the vulnerability detection of web applications as a whole.Therefore, the results of the previous work packages will be applicable to both the code run on a server and the code executed in a traditional web browser. In particular, applying the technique developed in WP 4 will be used to identify vulnerability and security patterns involving both server and client because Hop.js produces code that is executed on both client and server side.

Expected results: The DAJA tools will be available to both open-source and closed communities. The results expected from this projects are:

- Extension of the open source Moose system analysis platform by defining a metamodel to analyze JavaScript applications;
- Provisions of tools to help developers in the maintenance and evolution of JavaScript applications;
- Creation of an international network of researchers, including senior researchers, but more importantly, young researchers (PhD, recent doctors) from Argentina, Chile, and France;
- Scientific publications in the major conferences of the field, two years is short to expect publication of the project results in a journal, but we intend to submit such results by the end of the project.

[3] `https://www.owasp.org`

[4] `https://www.owasp.org/index.php/Category:OWASP_Top_Ten_Project`

B3. Schedule, with main execution stages

Working Package	Activity	Semesters			
		1	2	3	4
WP1	FAMIX extension to support JavaScript	X			
	Adaptation of existing tools of Moose to the new meta-model	X			
WP2	Adaptation of class complexity visual metaphor to code smells	X			
	Implementation of class complexity visual metaphor as a visual framework		X		
WP3	Definition of patterns of anomalous source code	X			
	Implementation of a tool to identify the patterns		X		
	Empirical analysis of the approach on web sites and open source projects			X	
WP4	Analysis of strategies to remove code anomalies			X	
	Application to security and vulnerable code			X	
	Evaluation of the approach on web sites and open source projects				X
WP5	AOT compilation	X	X	X	
	Static analyses for security			X	X
	Apply the results of WP2 and WP4 to multitier programming			X	X

B4. Contributions

All partners will participate in the five work packages either for developing it or use its results and assess their value:

- UNICEN will be leader on *WP 1 – Defining a meta-model*, but will involve the three partners experience.
- University of Chile will be leader on *WP 2 – Visualizing JavaScript applications using polymetric views* given their experience with software visualization.
- UNICEN will be leader on *WP 3 – Identifying anomalous code* given their expertise with the identification and prioritization of code anomalies.
- Indes/Inria will be leader on *WP 4 – Application to security and vulnerable code* given their expertise in security flaws.
- Indes/Inria will be leader on *WP 5 – Application to Hop.js* since Indes/Inria is the developer of Hop.js.
- At least two members, usually the three, are involved in each work package.

The scientific contributions of DAJA will be:

- Empirical analysis of the phenomenon of code anomalies, and in particular security and vulnerability, on JavaScript applications
- Definition of techniques to identify and remove code anomalies and vulnerabilities from JavaScript applications

We believe personal relationships are a fundamental asset in an international project where direct communication is hampered by distance and time difference. In the first stage of the project, integration will be based on the personal relationships that the members of University of Chile established with the two other partners during past projects: Alexandre Bergel (Chile) worked with Santiago Vidal (Argentina), they have several publications together [22, 21, 20], and Santiago Vidal visited University of Chile in different occasions. Tamara Rezk (Indes/team) has a long collaboration history with Argentinian [2, 3] and Chilean Universities [4, 7]. Alexandre Bergel and Manuel Serrano have been exchanging ideas since more than a decade on Web programming (although never formally collaborating on a well identified project).

B5. Regional Aspects

Interactions will be promoted by the use of the common platform on the three locations. This will guarantee that results can be easily and efficiently exchanged, and experiments can be faithfully replicated. Although there is already strong interaction between University of Chile members and INDES/Inria or UNICEN, the project will allow to create a new axis of cooperation between INDES/Inria and UNICEN.

B6. Institutions and CVs of coordinators

Institutions:

- Indes/Inria, has 5 permanents researchers and about 10 members. The team studies models for diffuse computing and develop languages for secure diffuse applications. Diffuse applications, of which Web 2.0 applications are a notable example, are the new applications emerging from the convergence of broad network accessibility, rich personal digital environment, and vast sources of information. Strong security guarantees are required for these applications, which intrinsically rely on sharing private information over networks of mutually distrustful nodes connected by unreliable media.

- The ISISTAN-CONICET institute at UNICEN has 17 permantent researchers and about 30 members. The main research topics of the institute are focused on the development of techniques and tools to support the design, evaluation and reuse of software designs.

- The Pleiad lab at University of Chile has 4 permanent researchers and about 13 members as of today. Pleiad works on many ways to better support software development at different levels, from programming languages to development environments, including tools to support program understanding (debuggers, profilers, visualizers). Pleiad also studies the practice of programming, through mining software repositories as well as user studies. Pleiad is part of the Department of Computer Science (DCC) of the University of Chile.

Manuel Serrano

1/ **Personal data**
Birth date: 17/05/1966
Professional address (with telephone and e-mail):
Inria-Sophia Antipolis,
2004 route des Lucioles - BP 93,
Sophia Antipolis Cedex 06902
Phone: +33 4 92 38 76 39
email: manuel.serrano@inria.fr
Current job title and size of the research group: Senior Scientist, Head of the Indes/Inria team, which has 5 permanent researchers and about 10 members as of today.

2/ **Highest obtained degree (with indication of place and date)**
HdR obtained at University of Nice in 2000.

3/ **Professional activity in the last 5 years**
Senior Scientist, Head of the Indes/Inria team.

4/ **Other duties/ positions**

- Head of Science of Inria Sophia.
- Reviewer for the main conferences in the field (ICFP, DLS, COORDINATION, ...).

5/ **Awards, fellowships and external recognition**

- Hop: best open source software ACM Multimedia 2007.

6/ **Ongoing funded research projects with dates, titles, sources of funding**
Projet RAPP (Robotics Application Store) [2012-2016] (European project), UCF (Ubiquitous Content Framework) [2014-2018] French FUI. Ajacs [2024-2018] French ANR.

7/ **Projects approved in the least 5 years**
RAPP, UCF, AJACS, PWD (French ANR).

8/ **Publications**

8.1 – **Highlight the most important publications related to the project theme**

- Luo, Z. and Fragoso Santos. J. and Almeida Matos, A. and Rezk, T. "Mashic Compiler: Mashup Sandboxing using Inter-frame Communication", in *Journal of Computer Security*, 2016
- Serrano, M. "A Multitier Debugger for Web Applications". In *Proceedings of the 10th WEBIST conference (WEBIST'14)*, Barcelona, Spain, Apr, 2014.
- Berry, G. and Serrano, M. "Hop and HipHop : Multitier Web Orchestration" in *Proceedings of the ICDCIT 2014 conference*, Feb, 2014, pp. 1–13.
- Serrano, M. and Berry, G., "Multitier Programming in Hop - A first step toward programming 21st-century applications", in *Communications of the ACM, 55(8)*, Aug, 2012
- Boudol, G. et al., "Reasoning about Web Applications: An Operational Semantics for HOP", in *ACM Transactions on Programming Languages and Systems (TOPLAS)*, 34(2), New York, NY, USA, 2012.
- Serrano, M. and Queinnec, C., in *A multi-tier semantics for Hop*, Higher Order and Symbolic Computation (HOSC), 23(4), 2012.

8.2 – **Publications in cooperation with the project partners**

9/ **Theses oriented and post-doctoral fellows supervised**

9.1 – **Finished/defended in the last 5 years**

- Johan Grande (Indes/Inria), PhD, 09/2011 to 09/2015
- Jose Santos (Indes/Inria), PhD, 09/2010 to 09/2014

9.2 – **Ongoing**

- Francis Dome, (Indes/Inria), PhD thesis, since 2015
- Colin Vidal, (Indes/Inria), PhD thesis, since 2015

Alexandre Bergel

1/ **Personal data**
Birth date: 23/03/1978
Professional address (with telephone and e-mail): Blanco Encalada 2120, Of 409, Santiago, Chile
Current job title and size of the research group: Assistant professor, Pleiad has 4 permanent researchers and about 13 members as of today

2/ **Highest obtained degree (with indication of place and date)**
PhD obtained at University of Bern in November 2005

3/ **Professional activity in the last 5 years**
Assistant professor at the University of Chile, researcher in Pleiad

4/ **Other duties/ positions**

5/ **Awards, fellowships and external recognition**:

- Distinguished Academic, delivered by the University of Chile (April 24, 2015) for the book *Deep into Pharo* (`http://deepintopharo.com`)
- Distinguished Reviewer Award, delivered by the IEEE Computer Society SANER Conference 2015
- First position at the 9th Innovation Technology Awards organized by ESUG'12 for our project titled "Object-Centric Debugging".
- Third position at the 9th Innovation Technology Awards organized by ESUG'12 for our project titled "Roassal".
- Best paper award for "Generics and Reverse Generics for Pharo", ICSOFT'12.
- Finalist Desafio Intel 2011.
- Ernst Denert Award 2006, for my PhD work on Classboxes.
- Best paper award for "Classboxes: A Minimal Module Model Supporting Local Rebinding", JMLC'03.

6/ **Ongoing funded research projects with dates, titles, sources of funding**

7/ **Projects approved in the least 5 years**

- Project Fondecyt, Project Equipe Associée, Project Corea STICAmSud with the University of La Plata and INRIA
- Project Fondecyt "ALMA: Analyses and Tools for Monitoring and Improving Software Quality"
- Project ECOSud with the University of Nice Sophia Antipolis "ASPIRE: Assess, Visualize and Reengineer Software Engineering Processes"

8/ **Publications**

8.1 – **Highlight the most important publications related to the project theme**

- Santiago A. Vidal, Alexandre Bergel, J. Andrs Daz-Pace, Claudia Marcos – "Over-exposed Classes in Java: An Empirical Study". *Computer Languages, Systems & Structures. 2016.*
- Juan Pablo Sandoval Alcocer, Alexandre Bergel, Marco Tulio Valente – "Learning From Source Code History to Identify Performance Failures.
  Proceedings of 7th ACM/SPEC International Conference on Performance Engineering (ICPE'16)
- Paloma Oliveira, Marco Tulio Valente, Alexandre Bergel, Alexander Serebrenik – "Validating Metric Thresholds with Developers - an early result-". *Proceedings of IEEE International Conference on Software Maintenance and Evolution, early research achievement (ICSME-ERA'15)*
- Leonardo Humberto Silva, Marco Tulio Valente, Nicolas Anquetil, Daniel Carlos Hovadick Flix, Alexandre Bergel, Anne Etien – "JSClassFinder: A Tool to Detect Class-like Structures in JavaScript". *Proceedings of Brazilian Conference on Software: Theory and Practice (CBSOFT'15)*

- Leonardo Humberto Silva, Miguel Ramos, Marco Tulio Valente, Alexandre Bergel, Nicolas Anquetil – "Does JavaScript Software Embrace Classes?". *Proceedings of 22nd IEEE International Conference on Software Analysis, Evolution, and Reengineering (IEEE SANER '15)*

8.2 – **Publications in cooperation with the project partners**

- Santiago A. Vidal, Alexandre Bergel, J. Andrs Daz-Pace, Claudia Marcos – "Over-exposed Classes in Java: An Empirical Study". *Computer Languages, Systems & Structures. 2016.*
- Santiago A. Vidal, Alexandre Bergel, Claudia Marcos, J. Andrs Daz-Pace – "Understanding and Addressing Exhibitionism in Java Empirical Research about Method Accessibility". *Empirical Software Engineering (EMSE), 2016*
- Santiago A. Vidal, Claudia Marcos, Alexandre Bergel, Gabriela Arevalo – "Memoization Aspects: a Case Study". *Proceedings of the 3rd International Workshop on Smalltalk Technologies (IWST'11)*

9/ **Theses oriented and post-doctoral fellows supervised**

9.1 – **Finished/defended in the last 5 years**

- Juan Pablo Sandoval, title "Horizontal Profiling: A Sampling Technique To Identify Performance Regressions", 2012 – 2016

9.2 – **Ongoing**

- Juraj Kubelka, title "Communication between developers", started 2013
- Miguel Campusano, title "Live Programming in Robotic", started 2013
- Jose Esteban, title "Assessing Energy Consumption in Java Applications", started 2017

Santiago Vidal

1/ **Personal data**
Birth date: 21/09/1984
Professional address (with telephone and e-mail): Campus Universitario, Tandil, Buenos Aires, Argentina. +5492494507504. svidal@exa.unicen.edu.ar
Current job title and size of the research group: Assistant researcher, ISISTAN has 17 permanent researchers and about 30 members as of today

2/ **Highest obtained degree (with indication of place and date)**
PhD obtained at UNICEN University of Tandil, Buenos Aires, Argentina in December 2013

3/ **Professional activity in the last 5 years**
Assistant researcher at ISISTAN-CONICET, UNICEN University.

4/ **Other duties/ positions**

5/ **Awards, fellowships and external recognition**:

- Best PhD work on the First Latin-American School on Software Engineering (ELA-ES), Rio de Janeiro, Brazil, 2013.

6/ **Ongoing funded research projects with dates, titles, sources of funding**
PICT Project "Refactoring of Software Systems driven by the Architecture" (PICT 2014-0855). September 2015-August 2017. Funded by Agencia Nacional de Promocin Cientfica y Tecnolgica
PDTS Project "A method centered on software architecture for product lines with tool support" (PDTS-217). June 2016 - May 2018. Funded by CONICET.

7/ **Projects approved in the least 5 years**

8/ **Publications**

8.1 – **Highlight the most important publications related to the project theme**

- Vidal, S. A., Bergel, A., Marcos, C., & Daz-Pace, J. A. "Understanding and addressing exhibitionism in Java empirical research about method accessibility". *Empirical Software Engineering, 1-34, 2016*
- Vidal, S. A., Bergel, A., Daz-Pace, J. A., & Marcos, C. "Over-exposed classes in Java". *Computer Languages, Systems & Structures. In press, 2016.*
- Vidal, S. A., Marcos, C., & Daz-Pace, J. A. "An approach to prioritize code smells for refactoring". *Automated Software Engineering, 2014.*
- Vidal, S. A., & Marcos, C. A. "Building an expert system to assist system refactorization". *Expert Systems with Applications, 39(3), 3810-3816. 2012*
- Vidal, S. A., & Marcos, C. A. "Toward automated refactoring of crosscutting concerns into aspects". *Journal of Systems and Software, 86(6), 1482-1497. 2013*

8.2 – **Publications in cooperation with the project partners**

- Vidal, S., Marcos, C., Bergel, A., Arvalo, G. "Memoization aspects: a case study". *In Proceedings of the International Workshop on Smalltalk Technologies (p. 6). ACM. 2011*
- Vidal, S. A., Bergel, A., Marcos, C., Daz-Pace, J. A. Understanding and addressing exhibitionism in Java empirical research about method accessibility. Empirical Software Engineering, 2015.
- Vidal, S. A., Bergel, A., Daz-Pace, J. A., Marcos, C. "Over-exposed classes in Java". *Computer Languages, Systems & Structures. In press. 2016*

9/ **Theses oriented and post-doctoral fellows supervised**

9.1 – **Finished/defended in the last 5 years**

9.2 – **Ongoing**

B7. Additional information

Santiago Vidal and Alexandre Bergel have been collaborating since 2011 on the topic of software analysis. No formal collaboration currently exists currently with Manuel Serrano. However, several discussions have been held on applying software assessment and maintenance techniques on JavaScript code. Applying these techniques to monitor the quality of the code produced by Hop.js is valuable because improving the security of JavaScript applications represents a major opportunity.

B8. International referees

Suggest names of at least 3 international referees to evaluate the project. These researchers should not be connected to people in the project.

1. Houari Sarahoui (`http://www.iro.umontreal.ca/~sahraouh`)
2. Alexander Serebrenik (`http://www.win.tue.nl/~aserebre`)
3. Robert Hirschfeld (`http://www.hirschfeld.org/`)

Names of referees who should not review this project in your opinion (optional)

1.
2.
3.

References

[1] J. P. S. Alcocer, A. Bergel, S. Ducasse, and M. Denker. Performance evolution blueprint: Understanding the impact of software evolution on performance. In A. Telea, A. Kerren, and A. Marcus, editors, *VISSOFT*, pages 1–9. IEEE, 2013.

[2] G. Barthe, P. R. D'Argenio, and T. Rezk. Secure information flow by self-composition. In *17th IEEE Computer Security Foundations Workshop, (CSFW-17 2004), 28-30 June 2004, Pacific Grove, CA, USA*, pages 100–114, 2004.

[3] G. Barthe, P. R. D'Argenio, and T. Rezk. Secure information flow by self-composition. *Mathematical Structures in Computer Science*, 21(6):1207–1252, 2011.

[4] G. Barthe, A. Hevia, Z. Luo, T. Rezk, and B. Warinschi. Robustness guarantees for anonymity. In *Proceedings of the 23rd IEEE Computer Security Foundations Symposium, CSF 2010, Edinburgh, United Kingdom, July 17-19, 2010*, pages 91–106, 2010.

[5] A. Bergel, S. Maass, S. Ducasse, and T. Gîrba. A domain-specific language for visualizing software dependencies as a graph. In *Proceedings of 2nd IEEE Working Conference on Software Visualization (VISSOFT NIER)*, 2014.

[6] L. Christophe, R. Stevens, C. D. Roover, and W. D. Meuter. Prevalence and maintenance of automated functional tests for web applications. In *Proceedings of 30th International Conference on Software Maintenance and Evolution (ICSME '14)*, 2014.

[7] R. Cruz, , T. Rezk, B. Serpette, and E. Tanter. Type abstraction for relaxed noninterference. In *European Conference on Object-Oriented Programming (ECOOP) 2017*.

[8] W. Cunningham. The wycash portfolio management system. *OOPS Messenger*, 4(2):29–30, 1993.

[9] S. Ducasse, T. Gîrba, and O. Nierstrasz. Moose: an agile reengineering environment. In *ACM SIGSOFT Software Engineering Notes*, volume 30, pages 99–102. ACM, 2005.

[10] S. Ducasse and M. Lanza. The Class Blueprint: Visually supporting the understanding of classes. *Transactions on Software Engineering (TSE)*, 31(1):75–90, Jan. 2005.

[11] A. M. Fard and A. Mesbah. Jsnose: Detecting javascript code smells. In *Source Code Analysis and Manipulation (SCAM), 2013 IEEE 13th International Working Conference on*, pages 116–125. IEEE, 2013.

[12] M. Fowler. *Refactoring: improving the design of existing code*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1999.

[13] L. Gong, M. Pradel, M. Sridharan, and K. Sen. Dlint: Dynamically checking bad coding practices in javascript. In *Proceedings of the 2015 International Symposium on Software Testing and Analysis*, pages 94–105. ACM, 2015.

[14] S. Guarnieri, M. Pistoia, O. Tripp, J. Dolby, S. Teilhet, and R. Berg. Saving the world wide web from vulnerable javascript. In *Proceedings of the 2011 International Symposium on Software Testing and Analysis*, ISSTA '11, pages 177–187, New York, NY, USA, 2011. ACM.

[15] S. H. Jensen, M. Madsen, and A. Møller. Modeling the html dom and browser api in static analysis of javascript web applications. In *Proceedings of the 19th ACM SIGSOFT symposium and the 13th European conference on Foundations of software engineering*, pages 59–69. ACM, 2011.

[16] M. Lanza and S. Ducasse. Polymetric views—a lightweight visual approach to reverse engineering. *Transactions on Software Engineering (TSE)*, 29(9):782–795, Sept. 2003.

[17] O. Nierstrasz, S. Ducasse, and T. Grba. The story of moose: an agile reengineering environment. *ACM SIGSOFT Software Engineering Notes*, 30(5):1–10, 2005.

[18] J. F. Santos, T. Jensen, T. Rezk, and A. Schmitt. Hybrid typing of secure information flow in a javascript-like language. In *Trustworthy Global Computing - 10th International Symposium, TGC*, 2015.

[19] S. Vidal, H. Vazquez, J. A. Diaz-Pace, C. Marcos, A. Garcia, and W. Oizumi. Jspirit: a flexible tool for the analysis of code smells. In *2015 34th International Conference of the Chilean Computer Science Society (SCCC)*, pages 1–6, Nov 2015.

[20] S. A. Vidal, A. Bergel, J. A. Díaz-Pace, and C. Marcos. Over-exposed classes in java: An empirical study. *Computer Languages, Systems & Structures*, 2016. In press.

[21] S. A. Vidal, A. Bergel, C. Marcos, and J. A. Díaz-Pace. Understanding and addressing exhibitionism in java empirical research about method accessibility. *Empirical Software Engineering*, pages 1–34, 2015.

[22] S. A. Vidal, C. Marcos, G. Arévalo, and A. Bergel. Memoization aspects: a case study. In *Proceedings of the 3rd International Workshop on Smalltalk Technologies (IWST'11), Collocated with ESUG*, 2011.

[23] S. A. Vidal, C. Marcos, and J. A. Díaz-Pace. An approach to prioritize code smells for refactoring. *Automated Software Engineering*, pages 1–32, 2014.

C. Project Budget

Project title: Title

Participating institutions: INDES/Inria (France); University of Chile (Chile); UNICEN (Argentina)

The STIC-AmSud program **funds travel expenses** (air tickets and *per diem*) to researchers in research missions and workshops.

C1. First year (2018)

Planned missions – Year 1

Researcher	Status (student, junior, senior)	Institution	Origin	Destination	Planned date	Duration (max. 30 days)	Estimated cost of the trip (€)	Estimate of total *per diem* (€)	Trip and Mission funding institution[5]	Mission objectives
M. Serrano	senior	INDES/Inria	Sophia	Santiago	March 2018	12 days	1,500	90 x 12 = 1,080	INDES/Inria	WP1
S. Vidal	senior	UNICEN	Tandil	Santiago	March 2018	12 days	400	90 x 12 = 1,080	UNICEN	WP1
A. Bergel	senior	UChile	Santiago	Tandil	June 2018	12 days	400	90 x 12 = 1,080	UChile	WP2
C. Vidal	junior	INDES/Inria	Sophia	Tandil	May 2018	15 days	1,500	90 x 15 = 1,350	INDES/Inria	WP1
A. Infante	student	UChile	Santiago	Sophia	August 2018	15 days	1,500	100 x 15 = 1,500	UChile	WP2
S. Vidal	senior	UNICEN	Tandil	Sophia	September 2018	15 days	1,500	100 x 15 = 1,500	UNICEN	WP3
T.Rezk	senior	INDES/Inria	Sophia	Santiago	August 2018	15 days	1,500	90 x 12 = 1,080	INDES/Inria	WP4

[5]Each institution will pay for the trip and per diem of its own researchers.

CONSOLIDATED BUDGET: Year 1

Funding requested to the STIC-AmSud Program
Estimated costs (€)

	A. Travel costs (air tickets)	B- Maintenance costs (per diem)	TOTAL
MAEDI France			
CNRS France			
INRIA France	4,500	3,500	8000
Institut Mines-Tlecom France			
MINCYT Argentina	1,900	2,580	4,480
CAPES Brazil			
CONICYT Chile	1,900	2,580	4,480
CONACYT Paraguay			
CONCYTEC Peru			
ANII Uruguay			
MPPEUCT Venezuela			
SENESCYT Ecuador			
COLCIENCIAS Colombia			
Total requested funding to STIC-AmSud			
Other funding[6]			
TOTAL	8,300	8,660	16,960

Do you have additional funding sources for this project[7]? (if so please specify the amount and source (s)).

[7]Reserved for CNRS researchers

C2. Second year (2019)

Second year funding depends on approval of intermediate progress report.

Planned missions – Year 2

Researcher	Status (student, junior, senior)	Institution	Origin	Destination	Planned date	Duration (max. 30 days)	Estimated cost of the trip (€)	Estimate of total *per diem* (€)	Trip and Mission funding institution[8]	Mission objectives
H. Vazquez	student	UNICEN	Tandil	Sophia	March 2019	15 days	1,500	100 x 15 = 1,500	UNICEN	WP3
J. Kubelka	student	UChile	Santiago	Tandil	May 2019	12 days	400	90 x 12 = 1,080	UChile	WP4
Francis Some	student	INDES/Inria	Sophia	Santiago	September 2019	12 days	1,500	90 x 12 = 1,080	INDES/Inria	WP5
C. Marcos	senior	UNICEN	Tandil	Santiago	September 2019	12 days	400	90 x 12 = 1,080	UNICEN	WP5
D. Orellana	student	UChile	Santiago	Sophia	August 2019	15 days	1,500	100 x 15 = 1,500	UChile	WP5
T.Rezk	senior	INDES/Inria	Sophia	Tandil	August 2019	15 days	1,500	90 x 12 = 1,080	INDES/Inria	WP4

[8]Each institution will pay for the trip and per diem of its own researchers.

CONSOLIDATED BUDGET: Year 2

Funding requested to the STIC-AmSud Program
Estimated costs (€)

	A. Travel costs (air tickets)	B- Maintenance costs (per diem)	TOTAL
MAEDI France			
CNRS France			
INRIA France	4,500	3,500	8,000
Institut Mines-Tlecom France			
MINCYT Argentina	1,900	2,580	4,480
CAPES Brazil			
CONICYT Chile	1,900	2,580	4,480
CONACYT Paraguay			
CONCYTEC Peru			
ANII Uruguay			
MPPEUCT Venezuela			
SENESCYT Ecuador			
COLCIENCIAS Colombia			
Total requested funding to STIC-AmSud			
Other funding[9]			
TOTAL	8,300	8,660	16,960

Do you have additional funding sources for this project[10]? (if so please specify the amount and source (s)).

[10]Reserved for CNRS researchers

C3. BUDGET TOTALS

	Year 1	**Year 2**	**Total**
Funding requested to MAEDI (France)			
Funding requested to INRIA (France)	8,000	8,000	16,000
Funding requested to CNRS (France)			
Funding requested to Institut Mines-Telecom (France)			
Funding requested to MINCYT (Argentina)	4,480	4,480	8,960
Funding requested to CAPES (Brazil)			
Funding requested to CONICYT (Chile)	4,480	4,480	8,960
Funding requested to CONACYT (Paraguay)			
Funding requested to CONCYTEC (Peru)			
Funding requested to ANII (Uruguay)			
Funding requested to SENESCYT (Ecuador)			
Funding requested to MPPEUCT (Venezuela)			
Funding requested to COLCIENCIAS (Colombia)			
Matching funds from the partners	(see note below)		
Other sources	0	0	0
TOTAL	16,960	16,960	33,920

Note: There will be PhD students in the three groups working on the project. There will also be travels to conferences paid by the respective groups on their funds, as well as computers for the students. These costs are not reported in the cost table.