



Fundação Empresa
Escola de Engenharia da
UFRGS



EDITAL DE CONTRATAÇÃO– 001/2021

Projeto:	PARAMETRIZACAO DA GEOMETRIA E ARQUITETURA DE RESERVATORIOS (Termo de Cooperação Petrobras 5900.0112658.19.9)
Objeto:	Contratação de profissionais autônomos: 1. Desenvolvedor de software enquadrado nível 2 (01 vaga) 2. Desenvolvedor de software nível 1 (01 vaga)
Requisitos de qualificação e experiência profissional:	1. Desenvolvedor de software enquadrado nível 2: 1.1. Formação em nível de graduação em Computação, Informática ou Engenharia da Computação. 1.2. Formação complementar em Informática ou Computação – pós-graduação lato ou stricto sensu – concluída ou em andamento. 1.3. Experiência profissional mínima de 4 anos como desenvolvedor de software ou analista, engenheiro ou arquiteto de software. 1.4. Experiência profissional ou formação acadêmica em modelagem conceitual: modelos de conhecimento, projetos em semântica da informação, recuperação de informação. 1.5. Conhecimento prévio na área de aplicação do projeto: Geologia de Petróleo, demonstrado no Curriculum Lattes através de participação em projetos, desenvolvimento de aplicativos ou experiência profissional. 1.6. Domínio oral e escrito da língua inglesa comprovado na entrevista de seleção. 1.7. Curriculum lattes atualizado no portal do CNPQ. 1.8. Certidão Negativa da Receita Federal 1.9. Certidão Negativa de Execução Patrimonial do domicílio do contratado; 1.10. Cadastro fiscal do ISS – Prefeitura de Porto Alegre. 2. Desenvolvedor de software nível 1: 2.1. Formação em nível técnico em Informática ou Computação. 2.2. Formação complementar em Informática ou Computação – graduação – concluída ou em andamento. 2.3. Experiência profissional mínima de 1 ano como desenvolvedor



	<p>de software.</p> <ol style="list-style-type: none">2.4. Experiência profissional ou conhecimento acadêmico no desenvolvimento de aplicações de computação gráfica;2.5. Conhecimento prévio na área de aplicação do projeto: Geologia de Petróleo, demonstrado no Curriculum Lattes através de participação em projetos, desenvolvimento de aplicativos ou experiência profissional.2.6. Domínio oral e escrito da língua inglesa comprovado na entrevista de seleção;2.7. Curriculum lattes atualizado no portal do CNPQ;2.8. Certidão Negativa da Receita Federal2.9. Certidão Negativa de Execução Patrimonial do domicílio do contratado;2.10. Cadastro fiscal do ISS – Prefeitura de Porto Alegre.
<p>Descrição do Serviço Prestado:</p>	<ol style="list-style-type: none">1. Desenvolvedor de software enquadrado nível 2:<ol style="list-style-type: none">1.1. Desenvolvimento dos aplicativos de software propostos pelos pesquisadores do projeto.1.2. Coordenação das atividades de desenvolvimento de software dos bolsistas do projeto e integração dos códigos em um aplicativo funcional.1.3. Gerenciamento dos repositórios de software e conteúdos gerados pelo projeto.1.4. Desenvolvimento e documentação dos modelos conceituais e de dados das aplicações do projeto.1.5. Desenvolvimento e atualização dos websites de divulgação do projeto.1.6. Produção de relatórios e apresentações técnicas.2. Desenvolvedor de software nível 1:<ol style="list-style-type: none">2.1. Desenvolvimento dos aplicativos de software propostos pelos pesquisadores do projeto.2.2. Gerenciamento da infraestrutura física de Informática do laboratório: servidores e configuração de rede.2.3. Desenvolvimento dos algoritmos de visualização gráfica e simulação visual do projeto.2.4. Desenvolvimento e atualização dos websites de divulgação do



Fundação Empresa
Escola de Engenharia da
UFRGS



	projeto. 2.5. Produção de relatórios e apresentações técnicas. Local de atuação: atividades serão desenvolvidas em modelo de trabalho remoto enquanto assim estiver determinado em Portaria pela UFRGS. Uma vez liberado o trabalho presencial, as atividades serão desenvolvidas no Instituto de Informática da UFRGS, Campus do Vale, Porto Alegre.
Nome do supervisor:	Joel Luis Carbonera
Período de execução do serviço:	30 meses (03/2021 até 09/2023)
Remuneração:	<ol style="list-style-type: none">1. Desenvolvedor de Software nível 2: R\$ 4.700,00 (bruto)2. Desenvolvedor de Software nível 1.: R\$ 3.750,00 (bruto)3. Fonte dos recursos financeiros: Termo de Cooperação Petrobras 5900.0112658.19.9
Forma de contratação:	Seleção Pública - Contratação CLT A contratação será rescindida ao final do período de execução do serviço, ficando adstrita ao período de vigência do Termo de Cooperação.

Cronograma

05/02/2021	Publicação do edital – www.feeng.ufrgs.br
08/02/2021 a 12/02/2021	Inscrições dos candidatos através do e-mail: projetos@inf.ufrgs.br , contendo o assunto “GEORESERVATORIO - EDITAL DE CONTRATAÇÃO - 001/2021.” O mail deve conter as seguintes informações: <ol style="list-style-type: none">1. <i>Nome Completo:</i>2. <i>Cartão UFRGS (se aplicável)</i>3. <i>Celular para contato.</i>4. <i>Mail alternativo para contato.</i>5. <i>Vaga de interesse: Desenvolvedor Nivel 1 () Desenvolvedor Nivel 2 ()</i>6. Link para curriculum lattes do candidato7. Texto de até uma página com



Fundação Empresa
Escola de Engenharia da
UFRGS



	informações complementares não constantes no Curriculum Lattes que demonstrem a adequação do candidato às vagas divulgadas: projetos, experiências internacionais, conhecimentos específicos e outras informações pertinentes.
17/02/2021 à 23/02/2021	Realização das entrevistas (Informações complementares no Anexo I)
01/03/2021	Divulgação do resultado no site www.feeng.ufrgs.br
02/03/2021 a 05/03/2021	Prazo para interposição de recursos através de preenchimento do formulário anexo e envio escaneado e assinado para o email – licitacoes_feeng@ufrgs.br



Fundação Empresa
Escola de Engenharia da
UFRGS



ANEXO I

Antes de se candidatar a vaga, o candidato deve buscar informações sobre o projeto Georeservatório

(<https://www.dropbox.com/s/xcwp3en79gerq4h/GeometriaReservShortVersion.pdf?dl=0>) e sua área de aplicação – Geologia de Petróleo – através das publicações recentes dos pesquisadores e avaliar seu interesse em trabalhar neste tema.

1. O processo seletivo será conduzido por uma comissão de docentes pesquisadores dos projetos.
 2. A seleção ocorrerá em duas fases. Na **primeira fase** serão analisados os currículos disponibilizados na Plataforma Lattes e documentos enviados. Os candidatos que atenderem aos requisitos mínimos de cada vaga serão pré-selecionados. O resultado da pré-seleção será divulgado na página da FEENG.
 3. A **segunda fase** contempla entrevista com os candidatos pré-selecionados onde serão verificadas as informações declaradas no curriculum e texto de informações complementares, o nível de inglês, e o conhecimento na área de aplicação do projeto (exploração e geologia de petróleo). Podem ser solicitados documentos comprobatórios das informações declaradas no curriculum, a critério da comissão. Durante as entrevistas, a comissão pode verificar (através de testes, perguntas ou documentos) o nível de competência informado pelo candidato na matriz de competências (Anexo II) e ajustar os valores informados pelo candidato, caso avalie que o valor informado não corresponda à avaliação realizada na entrevista.
- 3.1 Serão selecionados os candidatos da segunda fase com maior Nota Final na tabela abaixo. Candidatos com nota menor que 6 serão desclassificados desta seleção.

Critério	Avaliação do candidato	Nota do Candidato no Critério
Classificação predominante na matriz de competência (Anexo II) e originalmente publicada em https://sijinjoseph.com/programmer-competency-matrix/	0 a 3	
Conhecimento na área da Computação de atuação da função: Modelagem Conceitual ou Computação Gráfica	0 a 3	
Conhecimento na área de atuação do projeto – Geologia ou Exploração de petróleo	0 a 3	
Fluência em inglês	0 a 1	
Nota final do candidato	0 a 10	



ANEXO II

Programmer Competency Matrix

Note that the knowledge for each level is cumulative; being at level n implies that you also know everything from the levels lower than n.

Thanks to Sijin Joseph for the original version of this chart, available in <https://sijinjoseph.com/programmer-competency-matrix/>

Computer Science

	2 ⁿ (Level 0)	n ² (Level 1)	n (Level 2)	log(n) (Level 3)	Candidate classification and comments
data structures	Doesn't know the difference between Array and LinkedList	Able to explain and use Arrays, LinkedLists, Dictionaries etc in practical programming tasks	Knows space and time tradeoffs of the basic data structures, Arrays vs LinkedLists, Able to explain how hashables can be implemented and can handle collisions, Priority queues and ways to implement them etc.	Knowledge of advanced data structures like B-trees, binomial and fibonacci heaps, AVL/Red Black trees, Splay Trees, Skip Lists, tries etc.	
algorithms	Unable to find the average of numbers in an array (It's hard to believe but I've interviewed such candidates)	Basic sorting, searching and data structure traversal and retrieval algorithms	Tree, Graph, simple greedy and divide and conquer algorithms, is able to understand the relevance of the levels of this matrix.	Able to recognize and code dynamic programming solutions, good knowledge of graph algorithms, good knowledge of numerical computation algorithms, able to identify NP problems etc.	
systems programming	Doesn't know what a compiler, linker or interpreter is	Basic understanding of compilers, linker and interpreters. Understands what assembly code is and how things	Understands kernel mode vs. user mode, multi-threading, synchronization primitives and how they're implemented, able to read	Understands the entire programming stack, hardware (CPU + Memory + Cache + Interrupts + microcode), binary code, assembly, static and dynamic linking, compilation,	



		work at the hardware level. Some knowledge of virtual memory and paging.	assembly code. Understands how networks work, understanding of network protocols and socket level programming.	interpretation, JIT compilation, garbage collection, heap, stack, memory addressing...	
--	--	--	--	--	--

Software Engineering

	2 ⁿ (Level 0)	n ² (Level 1)	n (Level 2)	log(n) (Level 3)	
source code version control	Folder backups by date	VSS and beginning CVS/SVN user	Proficient in using CVS and SVN features. Knows how to branch and merge, use patches setup repository properties etc.	Knowledge of distributed VCS systems. Has tried out Bzr/Mercurial/Darcs/Git	
build automation	Only knows how to build from IDE	Knows how to build the system from the command line	Can setup a script to build the basic system	Can setup a script to build the system and also documentation, installers, generate release notes and tag the code in source control	
automated testing	Thinks that all testing is the job of the tester	Has written automated unit tests and comes up with good unit test cases for the code that is being written	Has written code in TDD manner	Understands and is able to setup automated functional, load/performance and UI tests	

Programming

	2 ⁿ (Level 0)	n ² (Level 1)	n (Level 2)	log(n) (Level 3)	
problem decomposition	Only straight line code with copy paste for reuse	Able to break up problem into multiple functions	Able to come up with reusable functions/objects that solve the	Use of appropriate data structures and algorithms and comes up with	



			overall problem	generic/object-oriented code that encapsulate aspects of the problem that are subject to change.	
systems decomposition	Not able to think above the level of a single file/class	Able to break up problem space and design solution as long as it is within the same platform/technology	Able to design systems that span multiple technologies/platforms.	Able to visualize and design complex systems with multiple product lines and integrations with external systems. Also should be able to design operations support systems like monitoring, reporting, fail overs etc.	
communication	Cannot express thoughts/ideas to peers. Poor spelling and grammar.	Peers can understand what is being said. Good spelling and grammar.	Is able to effectively communicate with peers	Able to understand and communicate thoughts/design/ideas/specs in a unambiguous manner and adjusts communication as per the context	
code organization within a file	no evidence of organization within a file	Methods are grouped logically or by accessibility	Code is grouped into regions and well commented with references to other source files	File has license header, summary, well commented, consistent white space usage. The file should look beautiful.	
code organization across files	No thought given to organizing code across files	Related files are grouped into a folder	Each physical file has a unique purpose, for e.g. one class definition, one feature implementation etc.	Code organization at a physical level closely matches design and looking at file names and folder distribution provides insights into design	
source tree organization	Everything in one folder	Basic separation of code into logical folders.	No circular dependencies, binaries, libs, docs, builds, third-party code	Physical layout of source tree matches logical hierarchy and organization. The directory names and	



			all organized into appropriate folders	organization provide insights into the design of the system.	
code readability	Mono-syllable names	Good names for files, variables, classes, methods etc.	No long functions, comments explaining unusual code, bug fixes, code assumptions	Code assumptions are verified using asserts, code flows naturally - no deep nesting of conditionals or methods	
defensive coding	Doesn't understand the concept	Checks all arguments and asserts critical assumptions in code	Makes sure to check return values and check for exceptions around code that can fail.	Has his own library to help with defensive coding, writes unit tests that simulate faults	
error handling	Only codes the happy case	Basic error handling around code that can throw exceptions/generate errors	Ensures that error/exceptions leave program in good state, resources, connections and memory is all cleaned up properly	Codes to detect possible exception before, maintain consistent exception handling strategy in all layers of code, come up with guidelines on exception handling for entire system.	
IDE	Mostly uses IDE for text editing	Knows their way around the interface, able to effectively use the IDE using menus.	Knows keyboard shortcuts for most used operations.	Has written custom macros	
API	Needs to look up the documentation frequently	Has the most frequently used APIs in memory	Vast and In-depth knowledge of the API	Has written libraries that sit on top of the API to simplify frequently used tasks and to fill in gaps in the API	
frameworks	Has not used any framework outside of the core platform	Has heard about but not used the popular frameworks	Has used more than one framework in a professional capacity and is	Author of framework	



		available for the platform.	well-versed with the idioms of the frameworks.		
requirements	Takes the given requirements and codes to spec	Come up with questions regarding missed cases in the spec	Understand complete picture and come up with entire areas that need to be speced	Able to suggest better alternatives and flows to given requirements based on experience	
scripting	No knowledge of scripting tools	Batch files/shell scripts	Perl/Python/Ruby/VBScript/PowerShell	Has written and published reusable code	
database	Thinks that Excel is a database	Knows basic database concepts, normalization, ACID, transactions and can write simple selects	Able to design good and normalized database schemas keeping in mind the queries that'll have to be run, proficient in use of views, stored procedures, triggers and user defined types. Knows difference between clustered and non-clustered indexes. Proficient in use of ORM tools.	Can do basic database administration, performance optimization, index optimization, write advanced select queries, able to replace cursor usage with relational sql, understands how data is stored internally, understands how indexes are stored internally, understands how databases can be mirrored, replicated etc. Understands how the two phase commit works.	

Experience

	2 ⁿ (Level 0)	n ² (Level 1)	n (Level 2)	log(n) (Level 3)	
languages with professional experience	Imperative or Object Oriented	Imperative, Object-Oriented and declarative (SQL), added bonus if they understand static vs	Functional, added bonus if they understand lazy evaluation, currying, continuations	Concurrent (Erlang, Oz) and Logic (Prolog)	



		dynamic typing, weak vs strong typing and static inferred types			
platforms with professional experience	1	2-3	4-5	6+	
years of professional experience	1	2-5	6-9	10+	
domain knowledge	No knowledge of the domain	Has worked on at least one product in the domain.	Has worked on multiple products in the same domain.	Domain expert. Has designed and implemented several products/solutions in the domain. Well versed with standard terms, protocols used in the domain.	

Knowledge

	2 ⁿ (Level 0)	n ² (Level 1)	n (Level 2)	log(n) (Level 3)	
tool knowledge	Limited to primary IDE (VS.Net, Eclipse etc.)	Knows about some alternatives to popular and standard tools.	Good knowledge of editors, debuggers, IDEs, open source alternatives etc. etc. For e.g. someone who knows most of the tools from Scott Hanselman's power tools list. Has used ORM tools.	Has actually written tools and scripts, added bonus if they've been published.	
languages exposed to	Imperative or Object Oriented	Imperative, Object-Oriented and	Functional, added bonus if they understand	Concurrent (Erlang, Oz) and Logic (Prolog)	



		declarative (SQL), added bonus if they understand static vs dynamic typing, weak vs strong typing and static inferred types	lazy evaluation, currying, continuations		
codebase knowledge	Has never looked at the codebase	Basic knowledge of the code layout and how to build the system	Good working knowledge of code base, has implemented several bug fixes and maybe some small features.	Has implemented multiple big features in the codebase and can easily visualize the changes required for most features or bug fixes.	
knowledge of upcoming technologies	Has not heard of the upcoming technologies	Has heard of upcoming technologies in the field	Has downloaded the alpha preview/CTP/beta and read some articles/manuals	Has played with the previews and has actually built something with it and as a bonus shared that with everyone else	
	2 ⁿ (Level 0)	n ² (Level 1)	n (Level 2)	log(n) (Level 3)	
platform internals	Zero knowledge of platform internals	Has basic knowledge of how the platform works internally	Deep knowledge of platform internals and can visualize how the platform takes the program and converts it into executable code.	Has written tools to enhance or provide information on platform internals. For e.g. disassemblers, decompilers, debuggers etc.	
books	Unleashed series, 21 days series, 24 hour series, dummies series...	Code Complete, Don't Make me Think, Mastering Regular Expressions	Design Patterns, Peopleware, Programming Pearls, Algorithm Design Manual, Pragmatic Programmer,	Structure and Interpretation of Computer Programs, Concepts Techniques, Models of Computer Programming, Art of Computer Programming,	



Fundação Empresa
Escola de Engenharia da
UFRGS



			Mythical Man month	Database systems , by C. J Date, Thinking Forth, Little Schemer	
blogs	Has heard of them but never got the time.	Reads tech/program ming/software engineering blogs and listens to podcasts regularly.	Maintains a link blog with some collection of useful articles and tools that he/she has collected	Maintains a blog in which personal insights and thoughts on programming are shared	



Fundação Empresa
Escola de Engenharia da
UFRGS



ANEXO III

Formulário para interposição de recurso relativo ao Edital 001/2021 para atuar no projeto “PARAMETRIZACAO DA GEOMETRIA E ARQUITETURA DE RESERVATORIOS”.

Eu,....., portador do documento de identidade nº.....para concorrer a uma vaga relativa a xxxxxxxxxxxxxxxx, a ser prestado para o projeto intitulado “**PARAMETRIZACAO DA GEOMETRIA E ARQUITETURA DE RESERVATORIOS**”, apresento o presente recurso:

A decisão objeto de contestação é.....

..... (explicitar a decisão que está contestando).

Os argumentos com os quais contesto a referida decisão são:

.....

.....

.....Para fundamentar essa contestação, encaminho anexos os seguintes documentos:

.....

.....

Porto Alegre ,.....de.....de 2021.

.....

Assinatura do candidato

RECEBIDO em...../...../...200.....

por.....

(Assinatura e cargo/função do servidor que receber o recurso)