

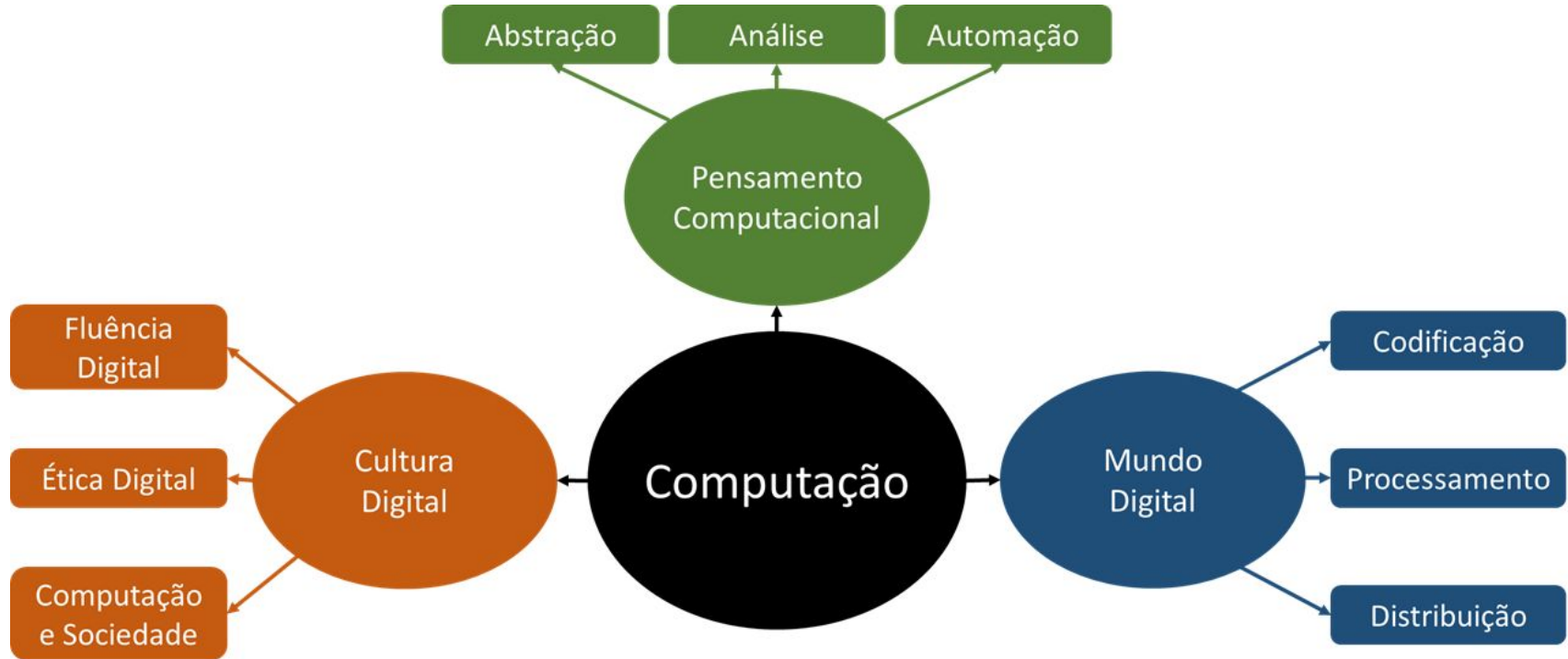
Uma Análise do Sequenciamento Pedagógico do Ensino de Computação na Educação Básica

**Nathalia da Cruz Alves, Christiane Gresse von Wangenheim,
Jean C. R. Hauck, Adriano F. Borgatto, Dalton F. de Andrade**



Ensino de computação na Educação Básica - SBC

Ensino de computação envolve o ensino do **pensamento computacional**



Diretrizes para o ensino do pensamento computacional no Ensino Fundamental

Ano	Objeto de conhecimento	Habilidades
1 Ano	Organização de objetos	Organizar objetos concretos de maneira lógica utilizando diferentes características (por exemplo: cor, tamanho, forma, texturas, detalhes, etc.).
	Algoritmo: definição	Compreender a necessidade de algoritmos para resolver problemas.
		Compreender a definição de algoritmos resolvendo problemas passo-a-passo (exemplos: construção de origamis, orientação espacial, execução de uma receita, etc.).
2 Ano	Identificação de padrões de comportamento	Identificar padrões de comportamento (exemplos: jogar jogos, rotinas do dia-a-dia, etc.).
	Algoritmos: construção e simulação	Definir e simular algoritmos (descritos em linguagem natural ou pictográfica) construídos como sequências e repetições simples de um conjunto de instruções básicas (avance, vire à direita, vire à esquerda, etc.).
...	...	...
9 Ano	Estruturas de dados: grafos e árvores	Formalizar os conceitos de grafo e árvore.
		Conhecer algoritmos básicos de tratamento das estruturas árvores e grafos.
	Técnica de construção de algoritmos: Generalização	Identificar problemas similares e a possibilidade do reuso de soluções, usando a técnica de generalização.
	Programação: generalização e grafos	Construir soluções de problemas usando a técnica de generalização, permitindo o reuso de soluções de problemas em outros contextos, aperfeiçoando e articulando saberes escolares.
		Identificar problemas de diversas áreas do conhecimento e criar soluções, de forma individual e colaborativa, através de programas de computador usando grafos e árvores.

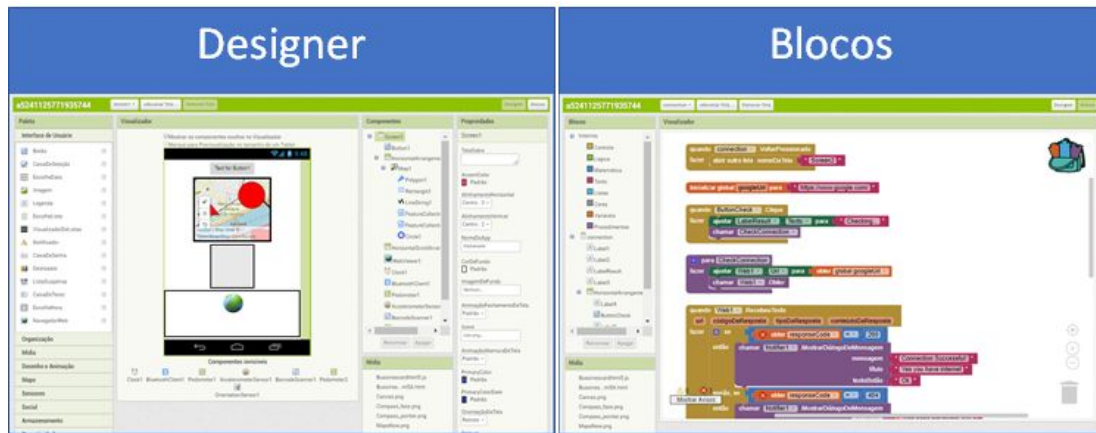
Ensinando programação

Ensino de alguns conceitos do pensamento computacional pode ser por meio do **desenvolvimento de aplicativos** com App Inventor

App Inventor é uma linguagem de programação visual que permite:

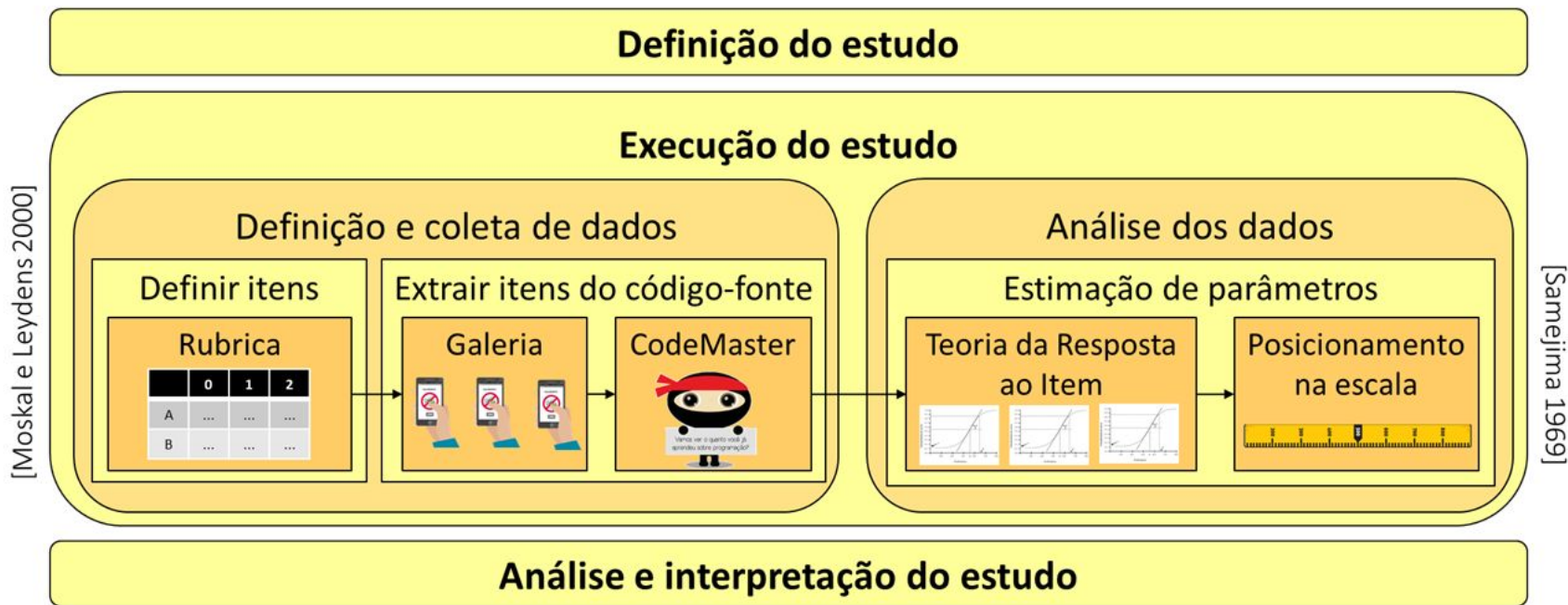
- **Criar aplicativos** para Android
- **Programar** encaixando blocos visuais de comandos
- **Visualizar** em tempo real o aplicativo em qualquer *smartphone* Android

Interface do App Inventor



Pergunta e método de pesquisa

O **sequenciamento** dos objetos de conhecimento das diretrizes da SBC que abordam **programação** por meio do eixo de pensamento computacional no Ensino Fundamental é adequado?



Rubrica CodeMaster para avaliar projetos App Inventor



Item	Insuficiente – 0 pontos	Básico – 1 ponto	Intermediário – 2 pontos	Avançado – 3 pontos
01. Eventos	Não usa eventos.	1 tipo de manipulador de eventos é usado.	2 tipos de manipuladores de eventos são usados.	Mais de 2 tipos de manipuladores de eventos são usados.
02. Variáveis	Não usa variáveis	Modificação ou uso de variáveis predefinidas.	Criação e operação com variáveis originais.	
03. Strings	Não usa strings.	Uso de string para alterar textos de componentes visuais.	Criação e operação com strings originais.	
04. Operadores	Não usa operadores.	Usa operadores aritméticos.	Usa operadores relacionais.	Usa operadores booleanos (lógicos).
05. Nomeação	Menos de 10% dos nomes são alterados.	10 a 25% dos nomes são alterados do padrão.	De 26 a 75% dos nomes são alterados do padrão.	Mais de 76% dos nomes são alterados do padrão.
06. Condicionais	Não usa condicionais.	Uso de condicional simples (if then).	Uso de condicional completo (if then else).	Uso de condicionais (if then else, if then).
07. Sincronização	Não usa temporizador para sincronização.	Uso de temporizador para sincronização.	-	-
08. Abstração	Não define procedimentos.	Existe exatamente um procedimento e sua chamada.	Existem procedimentos para organização (várias definições de procedimentos).	Existem procedimentos tanto para organização quanto para reuso (mais chamadas do que definição de procedimento)
09. Laços	Não usa laços.	Uso de laço simples (while)	Uso de laço com variável simples (For each)	Uso de laço com item de lista (For each).
10. Listas	Não usa listas.	Usa uma lista unidimensional.	Usa mais de uma lista unidimensional.	Usa uma lista de tuplas (map).
11. Persistência de dados	Não faz persistência de dados.	Usa persistência em arquivo.	Usa um banco de dados local.	Usa um banco de dados web.

Processo de avaliação via código-fonte

Avaliação do Pensamento Computacional

O professor define um **tema** em uma **atividade aberta**

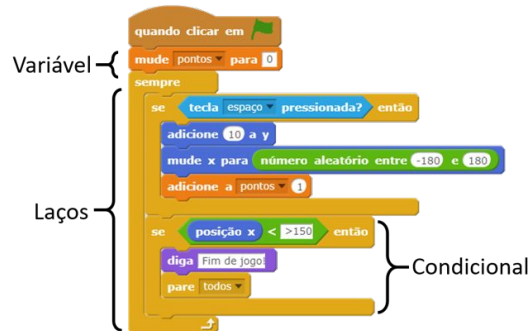
O aluno **programa** um aplicativo sobre o tema

O **código-fonte** do aplicativo criado pelo aluno é **analisado**

O aplicativo do aluno é **avaliado** com base nos dados da análise

Exemplo:

Conscientização do combate à Dengue



Critério	Pontuação
Variável	3
Laços	2
...	...
Nota final: 8	

Teoria da Resposta ao Item (TRI)

- Usada no ENEM, PISA, TOEFL, etc.
- Permite analisar várias características de um item
- **Modelo de Resposta Gradual** (proposto por Samejima)
 - Parâmetro **a** : inclinação
 - São esperados **valores acima de 1,0**
 - Parâmetros **b 's**: dificuldade
 - São esperados **valores dentro do intervalo [-5, 5]**

Calibração dos itens e posicionamento na escala

Parâmetros **a** valores acima de 1,0

Parâmetros **b's**: dentro do intervalo [-5, 5]

Item	a	EP(a)	b2	EP(b2)	b3	EP(b3)	b4	EP(b4)
I01. Eventos	2,877	0,022	-1,650	0,009	-0,902	0,006	-0,473	0,005
I02. Variáveis	2,971	0,022	-0,830	0,006	-0,009	0,005	-	-
I03. Strings	1,656	0,012	-0,568	0,007	0,942	0,008	-	-
I04. Operadores	3,081	0,024	-0,055	0,005	0,210	0,005	0,475	0,005
I05. Nomeação	1,680	0,012	-0,313	0,006	0,067	0,006	1,887	0,012
I06. Condicionais	2,323	0,017	0,344	0,005	0,796	0,006	1,571	0,009
I07. Sincronização	2,809	0,029	0,892	0,006	-	-	-	-
I08. Abstração	3,184	0,034	0,988	0,006	1,081	0,006	1,189	0,07
I09. Laços	1,766	0,027	2,138	0,021	2,293	0,023	2,575	0,027
I10. Listas	1,243	0,014	1,489	0,014	1,996	0,019	5,204	0,070
I11. Persistência	1,572	0,020	1,821	0,016	1,901	0,017	3,356	0,036

Calibração dos itens e posicionamento na escala

Parâmetros **a** valores acima de 1,0
Parâmetros **b's**: dentro do intervalo [-5, 5]

Pontos da
escala



Item	a	EP(a)	b2	EP(b2)	b3	EP(b3)	b4	EP(b4)
I01. Eventos	2,877	0,022	-1,650	0,009	-0,902	0,006	-0,473	0,005
I02. Variáveis	2,971	0,022	-0,830	0,006	-0,009	0,005	-	-
I03. Strings	1,656	0,012	-0,568	0,007	0,942	0,008	-	-
I04. Operadores	3,081	0,024	-0,055	0,005	0,210	0,005	0,475	0,005
I05. Nomeação	1,680	0,012	-0,313	0,006	0,067	0,006	1,887	0,012
I06. Condicionais	2,323	0,017	0,344	0,005	0,796	0,006	1,571	0,009
I07. Sincronização	2,809	0,029	0,892	0,006	-	-	-	-
I08. Abstração	3,184	0,034	0,988	0,006	1,081	0,006	1,189	0,07
I09. Laços	1,766	0,027	2,138	0,021	2,293	0,023	2,575	0,027
I10. Listas	1,243	0,014	1,489	0,014	1,996	0,019	5,204	0,070
I11. Persistência	1,572	0,020	1,821	0,016	1,901	0,017	3,356	0,036

Análise do sequenciamento dos objetos de conhecimento

Sequenciamento do eixo Pensamento Computacional das diretrizes da SBC no Ensino Fundamental		Sequenciamento baseado na TRI										
		← Mais fácil						Mais difícil →				
		I01.	I02.	I03.	I04.	I05.	I06.	I07.	I08.	I09.	I10.	I11.
		Eventos	Variáveis	Strings	Operadores	Nomeação	Condicionais	Sincronização	Abstração	Laços	Listas	Persistência
Mais fácil →	1 Ano	Organização de objetos										
		Algoritmo: definição										
	2 Ano	Identificação de padrões de comportamento										
		Modelos de objetos							X			
		Algoritmos: construção e simulação								X		
	3 Ano	Definição de problemas										
		Algoritmo: seleção			X		X			X		
		Introdução à lógica										
	4 Ano	Estruturas de dados estáticas: registros e vetores									X	
		Algoritmo: repetição								X		
← Mais difícil	5 Ano	Estruturas de dados dinâmicas: listas e grafos									X	
		Algoritmos sobre estruturas dinâmicas									X	
	6 Ano	Tipos de dados		X								
		Linguagem visual de programação					X			X		
		Introdução à generalização							X			
		Técnicas de solução de problema: decomposição							X			
	7 Ano	Automatização							X			
		Estruturas de dados: registros e vetores									X	
		Técnicas de solução de problemas: decomposição e reuso							X			
		Programação: decomposição e reuso							X			
	8 Ano	Estruturas de dados: listas									X	
		Programação: listas e recursão									X	
		Técnicas de solução de problema: recursão										
		Paralelismo						X				
	9 Ano	Estruturas de dados: grafos e árvores										
		Programação: generalização e grafos							X			
		Técnica de construção de algoritmos: Generalização							X			

Conclusões

- Diretrizes coerentes com as competências gerais da BNCC
- Dificuldade de aprendizagem observada na prática usando uma **linguagem de programação visual** não é refletida no sequenciamento dos objetos de conhecimento
- Base para sistematicamente discutir e melhorar o sequenciamento pedagógico das diretrizes
 - adotando técnicas de *scaffolding*
 - comparações com outros frameworks de referência

Obrigada! Perguntas?

Uma Análise do Sequenciamento Pedagógico no Ensino de Computação na Educação Básica

Assista também a apresentação do modelo de avaliação **CodeMaster** no dia **13/11/2019** às **8:50** na sala Mundo Novo 1 (CTD-IE-CBIE)



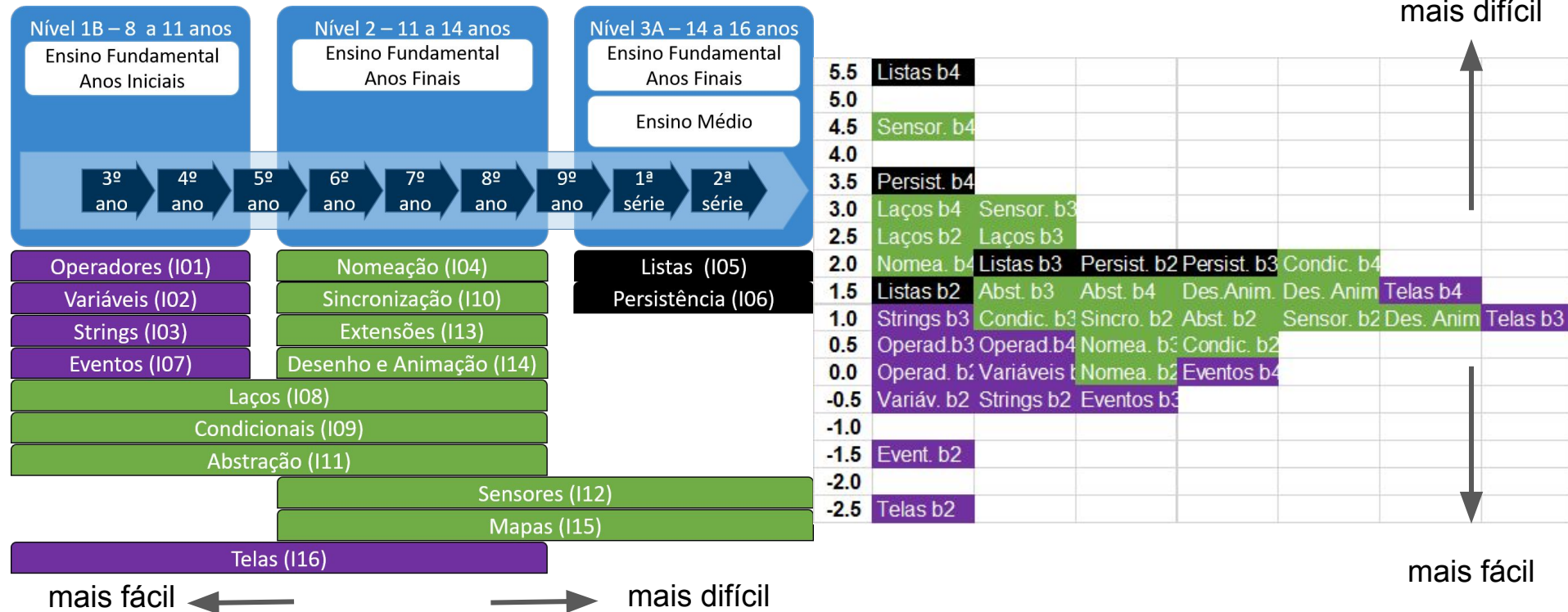
Nathalia da Cruz Alves

nathalia.alves@posgrad.ufsc.br

Slides extras

Resultados da avaliação do CodeMaster 2.0

Análise da **difículdade** dos itens usando o **Modelo de Resposta Gradual** da Teoria da Resposta ao Item, proposto por Samejima



CodeMaster

[http://apps.computacaonaescola.ufsc.br:8080/
rubrica_appinventor.jsp](http://apps.computacaonaescola.ufsc.br:8080/rubrica_appinventor.jsp)