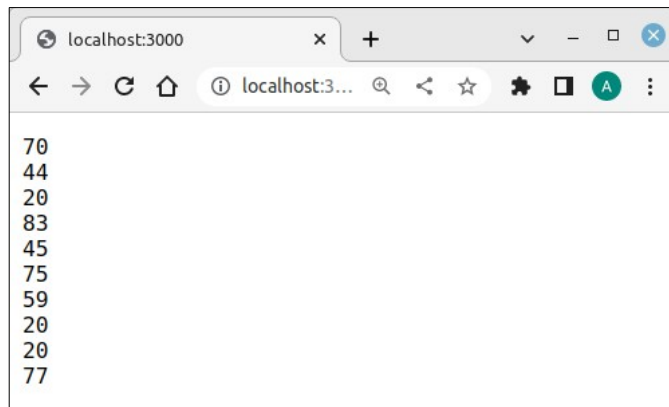
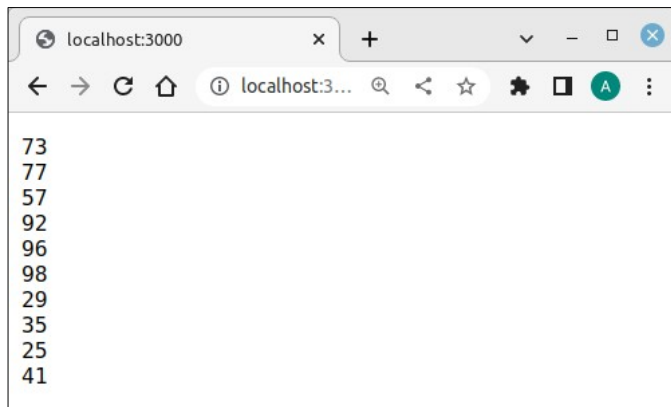
	<b>LISTA DE EXERCÍCIOS 06</b>	
	<b>CURSO:</b> Bacharelado em Sistemas de Informação	<b>MODALIDADE:</b> Ensino Superior
	<b>MÓDULO/SEMESTRE/SÉRIE:</b> 5º	<b>PERÍODO LETIVO:</b> 2025.1
	<b>DISCIPLINA:</b> Programação Web	<b>CLASSE:</b> 20251.5.119.1N
	<b>DOCENTE:</b> Alexandre dos Santos Silva	

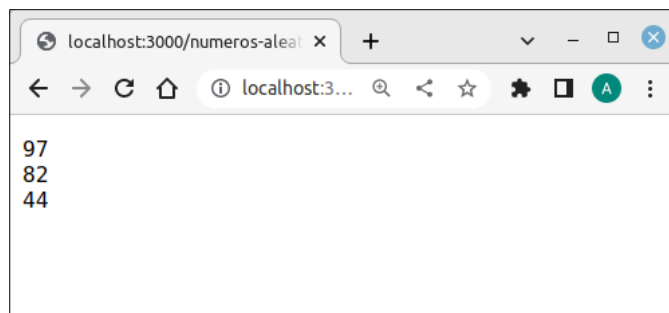
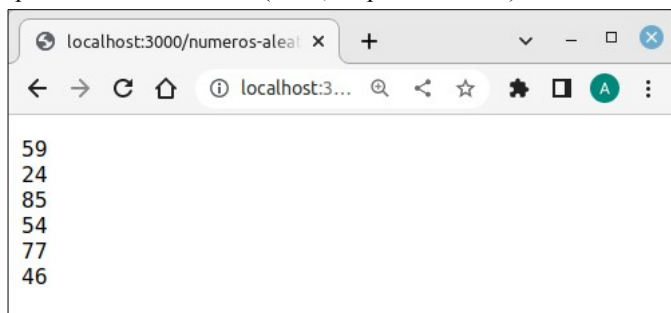
1. Implemente um **servidor web baseado em Node.js** no qual qualquer requisição HTTP leve à obtenção, na forma de texto simples, de uma sequência de 10 (dez) números inteiros pseudoaleatórios entre 1 (um) e (100), de modo que estejam separados por caracteres de quebra de linha. Para fins de exemplificação, na figura abaixo mostra-se resultado obtido com duas requisições em momentos distintos.



*Observação:* para a obtenção de cada número inteiro pseudoaleatório, considere o uso de funções estáticas disponibilizados pelo objeto global **Math**; para fins de demonstração, segue-se abaixo implementação de função com dois parâmetros com os quais se indica intervalo de número pseudoaleatório a ser gerado:

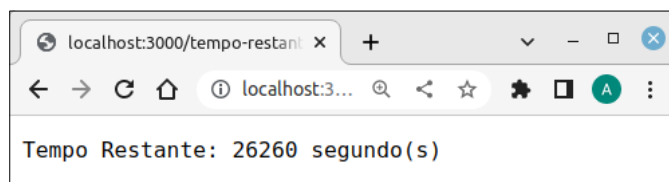
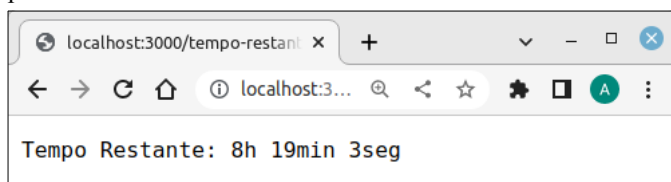
```
01 function getNumeroRandomico(min, max) {
02     min = Math.ceil(min); // maior inteiro inferior ao mínimo
03     max = Math.floor(max); // menor inteiro superior ao máximo
04     return Math.floor(Math.random() * (max - min)) + min; // retorno de número pseudoaleatório
05 }
```

2. Readeque o servidor web da questão anterior de modo que se possa indicar quantidade de números inteiros a serem gerados e retornados em resposta à requisição. Para o envio da quantidade de números na forma de *string* de consulta, considere método de requisição GET e rota específica (por exemplo, `/numeros-aleatorios`). Para fins de exemplificação, na figura abaixo são mostrados resultados obtidos com duas requisições que diferem entre si apenas em relação ao parâmetro de quantidade de números (3 e 6, respectivamente).

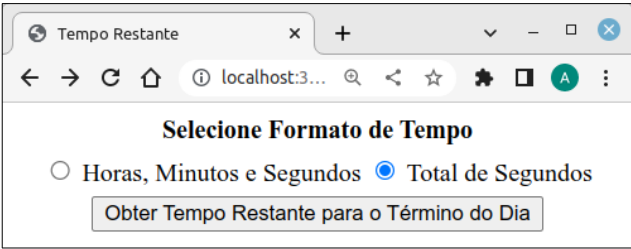


*Observação:* para a captura de parâmetro indicativo da quantidade de múltiplos, considere o uso do módulo **express**.

3. Implemente um **servidor web baseado em Node.js** que disponha da rota `/tempo-restante` para que se obtenha quantidade de tempo restante para o término do dia com base em horário corrente daquele servidor considerando-se requisição através por aquela rota. Quando do envio da requisição, deverá constar parâmetro na forma de *string* de consulta (utilizando-se, portanto, o método GET) para indicar formato de resposta da requisição HTTP (em horas, minutos e segundos ou em total de segundos). Para a captura do referido parâmetro, exige-se obrigatoriamente o uso do módulo **express**. Para fins de exemplificação, na figura abaixo são mostrados resultados obtidos com duas requisições que diferem entre si apenas em relação ao valor deste parâmetro.



Em tempo, considere adicionalmente que o servidor sirva formulário estático com aparência semelhante àquela mostrada na figura abaixo, de modo a permitir submissão de números através da rota de requisição aqui indicada; para tal, trate de usar *middleware* predefinido pela função `express.static`.

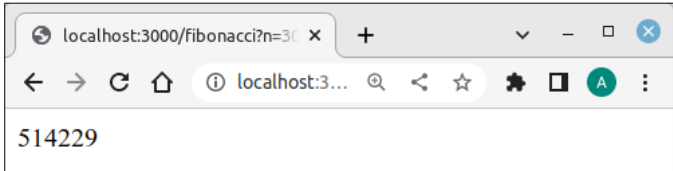
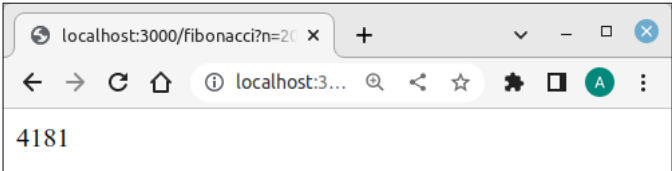


*Observação:* para a identificação do horário corrente, considere instanciação de objeto `Date` e de seus respectivos métodos para extração de horas, minutos e segundos. Por sua vez, em caso de eventual necessidade de obtenção apenas de valores inteiros de divisões (de modo que sejam ignorados valores decimais), considere invocação do método global `Math.ceil`.

4. A série de Fibonacci é uma sequência ilimitada de números inteiros em que seus dois primeiros termos são respectivamente 0 (zero) e 1 (um) e os termos subsequentes correspondem, cada um deles, à soma dos dois termos imediatamente anteriores. Deste modo, chega-se aos números que compõem a série (são listados apenas os primeiros 15 termos por se tratar de uma sequência ilimitada):

0,1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, 233, 377, ...

Implemente um **servidor web baseado em Node.js** que disponha da rota `/fibonacci` para se obter valor de enésimo termo da série de Fibonacci através de requisição por daquela rota; ao submeter a requisição, deverá constar parâmetro na forma de *string* de consulta (ou seja, via método GET) para indicar posição do termo da série. Para fins de exemplificação, na figura abaixo são mostrados resultados obtidos com duas requisições que diferem entre si apenas em relação ao valor deste parâmetro (20 e 30, respectivamente).



Em tempo, considere adicionalmente que o servidor sirva formulário estático com aparência semelhante àquela mostrada na figura abaixo, de modo a permitir submissão de posição de termo da série Fibonacci através da rota de requisição aqui indicada; para tal, trate de usar *middleware* predefinido pela função `express.static`.

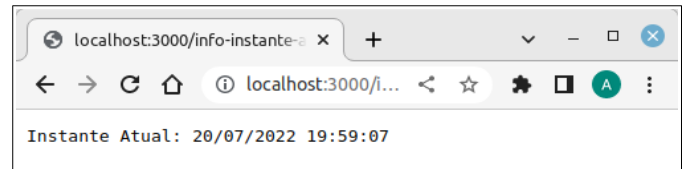
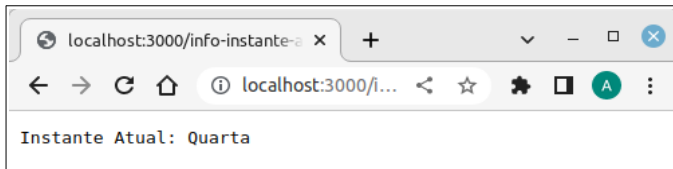


*Observação:* para o retorno do resultado numérico da operação aritmética na resposta de requisição, considere o formato JSON (JavaScript Object Notation) ou mesmo texto simples (neste último caso, considere o uso do método `toString`, que se aplica a qualquer objeto ou literal numérico). Por sua vez, para a obtenção de *string* em formato JSON correspondente a um objeto Java, sugere-se invocação de método estático `JSON.stringify`.

5. Implemente um **servidor web baseado em Node.js** que disponha da rota `/info-instante-atual` para que se obtenha alguma informação sobre a data e/ou horário corrente daquele servidor considerando-se requisição através por aquela rota. Quando do envio da requisição, deverá constar parâmetro na forma de *string* de consulta (utilizando-se, portanto, o método GET) para indicar informação de data e/ou horário corrente a ser devolvida em resposta à requisição HTTP; neste sentido, considere as seguintes informações (em alguns casos, combinação de duas ou mais dados da data e/ou horário corrente) acompanhadas de exemplo ilustrativo e, a título de sugestão, de identificador do parâmetro:

Informação	Exemplo	Parâmetro
Data (dia, ano e mês em formato numérico)	20/07/2022	<code>dma</code>
Nome do dia da semana	Quarta	<code>ds</code>
Data, horas, minutos e segundos	20/07/2022 19:30:25	<code>dmahms</code>
Horas, minutos e segundos	19:30:25	<code>hms</code>

Para a captura do referido parâmetro, exige-se obrigatoriamente o uso do módulo `express`. Para fins de exemplificação, na figura abaixo são mostrados resultados obtidos com duas requisições que diferem entre si apenas em relação ao valor deste parâmetro.



Em tempo, considere adicionalmente que o servidor sirva formulário estático com aparência semelhante àquela mostrada na figura abaixo, de modo a permitir submissão de informação de data e/ou horário corrente através da rota de requisição aqui indicada; para tal, trate de usar *middleware* predefinido pela função `express.static`.

A screenshot of a web browser window. The address bar shows 'localhost:3000'. The page content displays a form titled 'Selecione Formato de Exibição de Instante de Tempo Atual'. The form has four radio buttons: 'Data', 'Dia da Semana', 'Data, Horas, Minutos e Segundos', and 'Horas, Minutos e Segundos'. Below the radio buttons is a button labeled 'Obter Instante de Tempo Atual'.

*Observação:* para a identificação do horário corrente, considere instanciiação de objeto `Date` e de seus respectivos métodos para extração de dia do mês, dia da semana, mês, ano, horas, minutos e segundos.