APUNTES

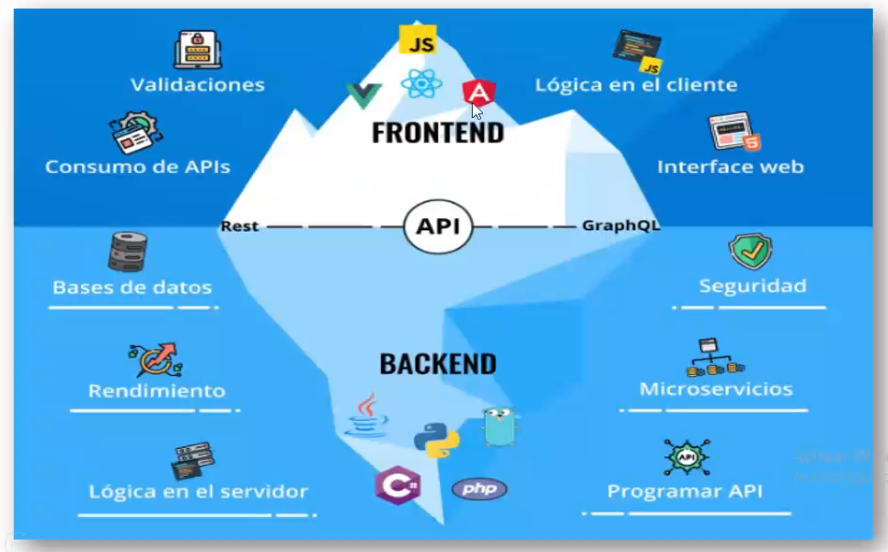
Instructor. CRISTIAN DARIO PIMIENTA: [cdpimientar@sena.edu.co](mailto:cdpimientar@sena.edu.co)

**SESIÓN UNO:**

1. Presentación de participantes
2. Ruta:



1. APP MONOLÍTIVA VS MICROSERVICIOS



1. Principios fundamentales de Dart

<https://dartpad.dev/>

|  |
| --- |
|  |

|  |
| --- |
| TALLER 1:   1. **Suma de dos números**: Escribe un programa que pida al usuario dos números enteros y luego muestre la suma de esos dos números. 2. **Factorial de un número**: Desarrolla un algoritmo que calcule el factorial de un número entero ingresado por el usuario. El factorial de un número n se denota como n! y se calcula como el producto de todos los enteros positivos desde 1 hasta n. 3. **Ordenar una lista**: Crea un programa que ordene una lista de números enteros en orden ascendente. Puedes implementar algún algoritmo de ordenamiento como el método de la burbuja o de selección. 4. **Buscar un elemento en una lista**: Escribe un algoritmo que busque un número dado en una lista de números enteros. Si el número está presente en la lista, el programa debe indicar su posición; de lo contrario, debe mostrar un mensaje indicando que el número no se encontró. 5. **Contar palabras en una cadena de texto**: Desarrolla un algoritmo que cuente el número de palabras en una cadena de texto ingresada por el usuario. Una palabra se define como una secuencia de caracteres separados por espacios en blanco. 6. **Calcular el promedio de una lista de números**: Escribe un programa que calcule el promedio de una lista de números enteros ingresados por el usuario. 7. **Convertir temperatura**: Crea un algoritmo que convierta una temperatura dada en grados Celsius a grados Fahrenheit. La fórmula de conversión es: Fahrenheit = (Celsius × 9/5) + 32. 8. **Encontrar el número más grande en una lista**: Desarrolla un programa que encuentre el número más grande en una lista de números enteros ingresados por el usuario. 9. **Fibonacci**: Escribe un algoritmo que genere los primeros n números de la secuencia de Fibonacci, donde cada número es la suma de los dos números anteriores. Los primeros dos números de la secuencia son 0 y 1. 10. **Verificar si un número es primo**: Crea un programa que verifique si un número dado por el usuario es primo o no. Un número primo es aquel que es divisible únicamente por 1 y por sí mismo. |

1. Jdklaf

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Ps** | **Descripción** | **Acción** |  | **Recursos** |
| **01** | Preparar entorno de trabajo (**terminal**). | django-admin startproject dj41\_practical\_structure\_models |  |  |
| **02** | Ir al proyecto (**terminal**). | cd dj41\_practical\_structure\_models |  |  |
| **03** | Crear una base de datos vacía en postgresql | agenda |  |  |
| **04** | Crear el módulo ‘contact’ (**terminal**). | python manage.py startapp contact |  |  |
| **05** | Crear el módulo ‘todo’ (**terminal**). | python manage.py startapp todo |  |  |
| **06** | Modificar el archivo ‘settings.py’ para reconocer el nuevo módulo. | [settings.py](https://github.com/ProfeAlbeiro/route02_python/blob/main/python04_projects/project05_django_inventory/dj41_practical_structure_models/dj41_practical_structure_models/settings.py). Líneas 40,41,80-85,112,114 |  |  |
| **07** | Verificar que el módulo ‘contact’ quedó instalado (**terminal**). | python manage.py check contact |  |  |
| **08** | Verificar que el módulo ‘todo’ quedó instalado (**terminal**). | python manage.py check todo |  |  |
| **09** | Modificar el archivo ‘url.py’ del módulo principal | [urls.py](https://github.com/ProfeAlbeiro/route02_python/blob/main/python04_projects/project05_django_inventory/dj41_practical_structure_models/dj41_practical_structure_models/urls.py) |  |  |
| **10** | Crear el archivo ‘urls.py’ del módulo ‘contact’ | [urls.py](https://github.com/ProfeAlbeiro/route02_python/blob/main/python04_projects/project05_django_inventory/dj41_practical_structure_models/contact/urls.py) |  |  |
| **11** | Crear el archivo ‘urls.py’ del módulo ‘todo’ | [urls.py](https://github.com/ProfeAlbeiro/route02_python/blob/main/python04_projects/project05_django_inventory/dj41_practical_structure_models/todo/urls.py) |  |  |
| **12** | Modificar el archivo ‘models.py’ del módulo ‘contact’ | [models.py](https://github.com/ProfeAlbeiro/route02_python/blob/main/python04_projects/project05_django_inventory/dj41_practical_structure_models/contact/models.py) |  |  |
| **13** | Modificar el archivo ‘models.py’ del módulo ‘todo’ | [models.py](https://github.com/ProfeAlbeiro/route02_python/blob/main/python04_projects/project05_django_inventory/dj41_practical_structure_models/todo/models.py) |  |  |
| **14** | Crear las tablas de ‘models.py’ a la base de datos (**terminal**). | python manage.py makemigrations |  |  |
| **15** | Actualizar la base de datos (**terminal**). | python manage.py migrate |  |  |
| **16** | Abrir el servidor de pruebas (**terminal**). | python manage.py runserver |  |  |
| **17** | Abrir la ruta del servidor en el navegador. | <http://127.0.0.1:8000> |  |  |
| **18** | Interrumpir el servidor (**terminal**). | ctrl + c |  |  |