Material descargable

egg

Soluciones Actividades

En este apartado, te compartimos algunas soluciones a los ejercicios que te hemos propuesto. Pero recuerda, no encontrarás un manual rígido ni una única manera de resolver los problemas. En su lugar, **te ofrecemos una perspectiva que abre puertas a diferentes maneras de enfrentar cada desafío.**

Cada actividad es una oportunidad para profundizar y comprender las posibles soluciones. Te animamos a ir más allá de buscar respuestas directas, y a utilizar tu curiosidad para explorar y personalizar los conocimientos adquiridos. Aquí, **el objetivo no es replicar respuestas, sino entender el proceso de pensamiento detrás de cada solución y cómo aplicarlo en distintas situaciones.**

Te alentamos a que, al utilizar estas soluciones, te tomes el tiempo necesario para comprender cada línea de código, para analizar cómo funciona y para adaptarlo a tus propias necesidades y proyectos.

La programación es un arte que requiere comprensión profunda y creatividad personal, y este espacio está diseñado para que desarrolles esas habilidades de manera óptima.

1. Crea un programa que genere un objeto LocalDate representando la fecha actual.

RESOLUCIÓN:

```
import java.time.LocalDate;

public class App {
  public static void main(String[] args) {
        // Obtener la fecha actual
        LocalDate fechaActual = LocalDate.now();

        // Imprimir la fecha actual
        System.out.println("La fecha actual es: " + fechaActual);
    }
}
```

2. Desarrolla un programa que genere un objeto LocalDate representando una fecha específica, como tu cumpleaños.

Material descargable

egg

Soluciones Actividades

RESOLUCIÓN:

```java
import java.time.LocalDate;

public class App {
  public static void main(String[] args) {
    // Especificar la fecha de tu cumpleaños
    LocalDate fechaCumpleaños = LocalDate.of(1986, 10, 16);

    // Imprimir la fecha de tu cumpleaños
    System.out.println("Mi cumpleaños es el: " + fechaCumpleaños);
  }
}
```

3. Implementa un programa que solicite al usuario ingresar por separado el día, el mes y el año, para luego convertir esta información en un objeto LocalDate.

RESOLUCIÓN:

```java
import java.time.LocalDate;
import java.util.Scanner;

public class App {
  public static void main(String[] args) {

    Scanner scanner = new Scanner(System.in);

    // Solicitar al usuario ingresar el día
    System.out.print("Ingrese el día: ");
    int dia = scanner.nextInt();

    // Solicitar al usuario ingresar el mes
    System.out.print("Ingrese el mes: ");
    int mes = scanner.nextInt();

    // Solicitar al usuario ingresar el año
    System.out.print("Ingrese el año: ");
    int año = scanner.nextInt();

    // Crear un objeto LocalDate con la información ingresada por el usuario
    LocalDate fechaIngresada = LocalDate.of(año, mes, dia);

    // Imprimir la fecha ingresada por el usuario
    System.out.println("La fecha ingresada es: " + fechaIngresada);

    scanner.close();
  }
}
```

Material descargable

egg

Soluciones Actividades

4. Escribe un programa que permita al usuario ingresar la fecha en un formato predeterminado por ti a través de la consola, y luego utilice esta información para crear un objeto LocalDate.

RESOLUCIÓN:

```
import java.time.LocalDate;
import java.time.format.DateTimeFormatter;
import java.time.format.DateTimeParseException;
import java.util.Scanner;

public class App {
  public static void main(String[] args) {
    Scanner scanner = new Scanner(System.in);

    // Definir el formato de fecha predeterminado
                              DateTimeFormatter       formatoFecha       =
DateTimeFormatter.ofPattern("dd/MM/yyyy");

    try {
      // Solicitar al usuario ingresar la fecha en el formato predeterminado
      System.out.print("Ingrese la fecha (dd/MM/yyyy): ");
      String fechaTexto = scanner.nextLine();

       // Convertir la cadena de texto a un objeto LocalDate utilizando el
formato
      // predeterminado
      LocalDate fechaIngresada = LocalDate.parse(fechaTexto, formatoFecha);

      // Imprimir la fecha ingresada por el usuario
      System.out.println("La fecha ingresada es: " + fechaIngresada);
    } catch (DateTimeParseException e) {
      // Manejar la excepción si el formato de fecha es incorrecto
        System.out.println("Error: Formato de fecha incorrecto. Por favor,
ingrese la fecha en el formato dd/MM/yyyy.");
    }

    scanner.close();
  }
}
```

5. Desarrolla un programa que solicite al usuario ingresar una fecha en formato 'dd-MM-yyyy'. Luego, el programa debe generar un objeto LocalDate basado en la información proporcionada. Posteriormente, añade 15 días, 2 meses y 3 años a esta fecha. Por último, muestra al usuario la nueva fecha obtenida y el día de la semana correspondiente.

Material descargable

egg

Soluciones Actividades

RESOLUCIÓN:

```
import java.time.DayOfWeek;
import java.time.LocalDate;
import java.time.format.DateTimeFormatter;
import java.util.Scanner;

public class App {
  public static void main(String[] args) {
    Scanner scanner = new Scanner(System.in);

    // Definir el formato de fecha
                                DateTimeFormatter        formatoFecha        =
DateTimeFormatter.ofPattern("dd-MM-yyyy");

    // Solicitar al usuario ingresar una fecha en formato 'dd-MM-yyyy'
    System.out.print("Ingrese una fecha en formato dd-MM-yyyy: ");
    String fechaTexto = scanner.nextLine();

      // Convertir la cadena de texto a un objeto LocalDate utilizando el
formato
    // especificado
    LocalDate fechaIngresada = LocalDate.parse(fechaTexto, formatoFecha);

    // Añadir 15 días, 2 meses y 3 años a la fecha ingresada
                                      LocalDate         nuevaFecha          =
fechaIngresada.plusDays(15).plusMonths(2).plusYears(3);

    // Obtener el día de la semana correspondiente a la nueva fecha
    DayOfWeek diaSemana = nuevaFecha.getDayOfWeek();

    // Imprimir la nueva fecha y el día de la semana correspondiente
    System.out.println("La nueva fecha es: " + nuevaFecha);
       System.out.println("El  día  de  la  semana  correspondiente  es:  "  +
diaSemana);

    scanner.close();
  }
}
```

6. Desarrolla un programa que solicite al usuario ingresar una fecha en formato 'dd-MM-yyyy'. Luego, el programa debe generar un objeto LocalDate basado en la información proporcionada. Posteriormente, resta 13 días, 10 meses y 1 año. a esta fecha. Por último, muestra al usuario la nueva fecha obtenida y el día de la semana correspondiente.

Material descargable

egg

Soluciones Actividades

RESOLUCIÓN:

```
import java.time.DayOfWeek;
import java.time.LocalDate;
import java.time.format.DateTimeFormatter;
import java.util.Scanner;

public class App {
  public static void main(String[] args) {
    Scanner scanner = new Scanner(System.in);

    // Definir el formato de fecha
                                DateTimeFormatter        formatoFecha        =
DateTimeFormatter.ofPattern("dd-MM-yyyy");

    // Solicitar al usuario ingresar una fecha en formato 'dd-MM-yyyy'
    System.out.print("Ingrese una fecha en formato dd-MM-yyyy: ");
    String fechaTexto = scanner.nextLine();

      // Convertir la cadena de texto a un objeto LocalDate utilizando el
formato especificado
    LocalDate fechaIngresada = LocalDate.parse(fechaTexto, formatoFecha);

    // Restar 13 días, 10 meses y 1 año a la fecha ingresada
                                      LocalDate         nuevaFecha          =
fechaIngresada.minusDays(13).minusMonths(10).minusYears(1);

    // Obtener el día de la semana correspondiente a la nueva fecha
    DayOfWeek diaSemana = nuevaFecha.getDayOfWeek();

    // Imprimir la nueva fecha y el día de la semana correspondiente
    System.out.println("La nueva fecha es: " + nuevaFecha);
       System.out.println("El  día  de  la  semana  correspondiente  es:  "  +
diaSemana);

    scanner.close();
}
}
```

7. Dado un objeto LocalDate ingresado por el usuario, tu tarea es verificar si el año correspondiente es un año bisiesto o no.

Material descargable

egg

Soluciones Actividades

RESOLUCIÓN:

```java
import java.time.LocalDate;
import java.util.Scanner;

public class App {
  public static void main(String[] args) {
    Scanner scanner = new Scanner(System.in);

    int dia, mes, año;
    LocalDate fechaIngresada = null;

    do {
      System.out.print("Ingrese el día (1-31): ");
      dia = scanner.nextInt();
    } while (dia < 1 || dia > 31);

    do {
      System.out.print("Ingrese el mes (1-12): ");
      mes = scanner.nextInt();
    } while (mes < 1 || mes > 12);

    do {
      System.out.print("Ingrese el año (YYYY): ");
      año = scanner.nextInt();
    } while (año < 0);

    // Verificar si la fecha es válida
    try {
      fechaIngresada = LocalDate.of(año, mes, dia);
    } catch (Exception e) {
      System.out.println("Fecha ingresada no válida.");
      System.exit(1);
    }

    // Verificar si el año es bisiesto o no
    if (fechaIngresada.isLeapYear()) {
      System.out.println("El año " + año + " es un año bisiesto.");
    } else {
      System.out.println("El año " + año + " no es un año bisiesto.");
    }

    scanner.close();
  }
}
```

Material descargable

egg

Soluciones Actividades

8. Diseña un juego de adivinanza de fechas, donde crearás una LocalDate aleatoria y le preguntarás al usuario cuál es. Solo podrás decirle si la fecha ingresada es antes o después de la fecha objetivo. Cuando adivine la fecha, el juego terminará y lo felicitarás por haberlo logrado

RESOLUCIÓN:

```
import java.time.LocalDate;
import java.time.format.DateTimeFormatter;
import java.util.Random;
import java.util.Scanner;

public class App {
  public static void main(String[] args) {
    Scanner scanner = new Scanner(System.in);
    Random random = new Random();
                                DateTimeFormatter        formatoFecha        =
DateTimeFormatter.ofPattern("dd-MM-yyyy");

    // Generar una fecha aleatoria
    LocalDate fechaObjetivo = LocalDate.of(
        random.nextInt(2022) + 1, // Año aleatorio entre 1 y 2022
        random.nextInt(12) + 1, // Mes aleatorio entre 1 y 12
            random.nextInt(28)  +  1  //  Día  aleatorio  entre  1  y  28  (para
simplificar)

    );

    System.out.println("Bienvenido al juego de adivinanza de fechas!");
        System.out.println("Se  ha  generado  una  fecha  aleatoria.  Intenta
adivinarla.");
       System.out.println("(IMPRIMIMOS  LA  FECHA  RANDOM,  PARA  TENERLA  DE
REFERENCIA)  " + fechaObjetivo);

    boolean adivinado = false;
    while (!adivinado) {
      System.out.print("Ingresa tu fecha (dd-MM-yyyy): ");
      String entradaUsuario = scanner.nextLine();

      // Convertir la entrada del usuario a LocalDate
              LocalDate   fechaIngresada   =   LocalDate.parse(entradaUsuario,
formatoFecha);

      // Comparar la fecha ingresada con la fecha objetivo
      int comparacion = fechaIngresada.compareTo(fechaObjetivo);

      if (comparacion == 0) {
                System.out.println("¡Felicidades!  Has  adivinado  la  fecha
correctamente.");
        adivinado = true;
      } else if (comparacion < 0) {
```

Soluciones Actividades

```
            System.out.println("La fecha ingresada es antes de la fecha
objetivo. Intenta nuevamente.");
        } else {
            System.out.println("La fecha ingresada es después de la fecha
objetivo. Intenta nuevamente.");
        }
    }

    scanner.close();
  }
}
```

9. Crea un objeto LocalTime representando la hora actual y obtén la cantidad de segundos que han pasado desde el inicio del día.

RESOLUCIÓN:

```
import java.time.LocalTime;

public class App {
  public static void main(String[] args) {
        // Crear un objeto LocalTime representando la hora actual
        LocalTime horaActual = LocalTime.now();

        // Obtener la cantidad de segundos desde el inicio del día
        long segundosDesdeInicioDelDia = horaActual.toSecondOfDay();

        // Imprimir la cantidad de segundos
        System.out.println("Cantidad de segundos desde el inicio del día: "
+ segundosDesdeInicioDelDia);
    }
 }
```

10. Crea un objeto LocalTime que represente una hora específica. Después, utiliza ese objeto para calcular cuántas horas, minutos y segundos faltan hasta la medianoche.

RESOLUCIÓN:

```
        // Crear un objeto LocalTime que represente una hora específica
        LocalTime horaActual = LocalTime.now();

            // Calcular cuántas horas, minutos y segundos faltan hasta la
medianoche
```

Material descargable

egg

Soluciones Actividades

```
        LocalTime medianoche = LocalTime.of(23, 59, 59); // Último segundo
del día
                long horasRestantes = ChronoUnit.HOURS.between(horaActual,
medianoche);
            long minutosRestantes = ChronoUnit.MINUTES.between(horaActual,
medianoche);
           long segundosRestantes = ChronoUnit.SECONDS.between(horaActual,
medianoche);

        // Mostrar el resultado
                System.out.println("Horas restantes hasta medianoche: " +
horasRestantes);
             System.out.println("Minutos restantes hasta medianoche: " +
minutosRestantes);
            System.out.println("Segundos restantes hasta medianoche: " +
segundosRestantes);
    }
}
```

11. Dado un objeto LocalTime ingresado por el usuario, ajusta la hora al próximo valor exacto.

RESOLUCIÓN:

```
import java.time.LocalTime;
import java.util.Scanner;

public class App {
  public static void main(String[] args) {
    Scanner scanner = new Scanner(System.in);

    // Solicitar al usuario que ingrese la hora
    System.out.print("Ingrese la hora (en formato HH:MM): ");
    String inputHora = scanner.nextLine();

    // Convertir la entrada del usuario a LocalTime
    LocalTime horaIngresada = LocalTime.parse(inputHora);

    // Obtener la hora siguiente en punto
                                LocalTime proximaHoraEnPunto =
horaIngresada.plusHours(1).withMinute(0).withSecond(0);

    // Mostrar el resultado
    System.out.println("El próximo valor exacto es: " + proximaHoraEnPunto);

    scanner.close();
}
}
```

Material descargable

egg

Soluciones Actividades

12. Dada una cantidad de segundos ingresada por el usuario, conviértela en un objeto LocalTime y muéstrala por consola.

RESOLUCIÓN:

```
import java.time.LocalTime;
import java.util.Scanner;

public class App {
  public static void main(String[] args) {
    Scanner scanner = new Scanner(System.in);

    // Solicitar al usuario que ingrese la cantidad de segundos
    System.out.print("Ingrese la cantidad de segundos: ");
    int segundos = scanner.nextInt();

    // Calcular horas, minutos y segundos
    int horas = segundos / 3600;
    int minutos = (segundos % 3600) / 60;
    int segundosRestantes = segundos % 60;

    // Crear un objeto LocalTime con los valores calculados
    LocalTime tiempo = LocalTime.of(horas, minutos, segundosRestantes);

     // Mostrar el resultado, la prmer linea en formato Local Time, segunda
linea en
    // datos independientes
    System.out.println("El tiempo equivalente es: " + tiempo);
      System.out.printf("El tiempo equivalente es: %d horas, %d minutos, %d
segundos\n", horas, minutos,
          segundosRestantes);
    scanner.close();
  }
}
```

13. Escribe un programa que calcule la diferencia en segundos entre dos objetos LocalTime.

Material descargable

egg

Soluciones Actividades

RESOLUCIÓN:

```
import java.time.LocalTime;
import java.time.temporal.ChronoUnit;


public class App {
  public static void main(String[] args) {
    // Crear dos objetos LocalTime
    LocalTime tiempo1 = LocalTime.of(10, 15, 30);
    LocalTime tiempo2 = LocalTime.of(13, 20, 40);

    // Calcular la diferencia en segundos
         long  diferenciaEnSegundos  =  ChronoUnit.SECONDS.between(tiempo1,
tiempo2);

    // Mostrar el resultado
     System.out.println("La diferencia en segundos entre los tiempos es: " +
diferenciaEnSegundos + " segundos.");
  }
}
```

14. Dado un objeto LocalTime que has ingresado, crea un nuevo LocalTime que represente exactamente la mitad del tiempo transcurrido entre la hora dada y la medianoche.

RESOLUCIÓN:

```
import java.time.LocalTime;
import java.time.temporal.ChronoUnit;

public class App {
  public static void main(String[] args) {
    // Crear un objeto LocalTime
     LocalTime horaDada = LocalTime.of(12, 30, 15); // Por ejemplo, 12:30:15
PM

    // Calcular la cantidad de segundos transcurridos desde la medianoche
                                  long        segundosDesdeMedianoche        =
ChronoUnit.SECONDS.between(LocalTime.MIDNIGHT, horaDada);

    // Calcular la mitad del tiempo
    long mitadSegundos = segundosDesdeMedianoche / 2;
        //  Crear  un  nuevo  LocalTime  representando  la  mitad  del  tiempo
transcurrido
    LocalTime mitadTiempo = LocalTime.MIDNIGHT.plusSeconds(mitadSegundos);
    // Mostrar el nuevo LocalTime
      System.out.println("La  mitad  del  tiempo  entre  la  medianoche  y  " +
horaDada + " es: " + mitadTiempo);
  }
}
```

Material descargable

egg

Soluciones Actividades

15. Crea un método que, dado un LocalDateTime, devuelva un ZonedDateTime que represente el mismo instante en la zona horaria de Tokio (Asia/Tokyo).

RESOLUCIÓN:

```
import java.time.ZoneId;
import java.time.ZonedDateTime;
import java.time.format.DateTimeFormatter;

public class App {
  public static void main(String[] args) {
    // Ejemplo de uso del método
    ZonedDateTime zonedDateTimeTokio = obtenerZonedDateTimeTokio();

    // Imprimir el resultado formateado
          imprimirZonedDateTime("La fecha y hora en Tokio es: ",
zonedDateTimeTokio);
  }

  // Método para obtener un ZonedDateTime en la zona horaria de Tokio
  public static ZonedDateTime obtenerZonedDateTimeTokio() {
    // Obtener la fecha y la hora actual
    return ZonedDateTime.now(ZoneId.of("Asia/Tokyo"));
  }

  // Método para imprimir un ZonedDateTime formateado
   public static void imprimirZonedDateTime(String mensaje, ZonedDateTime
zonedDateTime) {
    // Formatear el ZonedDateTime para una salida legible
      DateTimeFormatter formatter = DateTimeFormatter.ofPattern("yyyy-MM-dd
HH:mm:ss z");
    String formattedDateTime = zonedDateTime.format(formatter);

    // Imprimir el mensaje y el ZonedDateTime formateado
    System.out.println(mensaje + formattedDateTime);
  }
}
```

16. Solicita al usuario una fecha y hora y crea un ZonedDateTime para luego mostrarle que fecha y hora seria en todas las zonas horarias.

RESOLUCIÓN:

```
import java.time.LocalDateTime;
import java.time.ZoneId;
import java.time.ZonedDateTime;
import java.time.format.DateTimeFormatter;
import java.util.Scanner;
```

Material descargable

egg

Soluciones Actividades

```
import java.util.Set;

public class App {
  public static void main(String[] args) {
    // Solicitar al usuario que ingrese la fecha y la hora
    Scanner scanner = new Scanner(System.in);
    System.out.println("Ingrese la fecha y hora (yyyy-MM-dd HH:mm:ss):");
    String fechaHoraTexto = scanner.nextLine();

    // Convertir la entrada del usuario a LocalDateTime
          LocalDateTime   fechaHora   =   LocalDateTime.parse(fechaHoraTexto,
DateTimeFormatter.ofPattern("yyyy-MM-dd HH:mm:ss"));

    // Crear un ZonedDateTime para la zona horaria del sistema
            ZonedDateTime   zonedDateTime   =   ZonedDateTime.of(fechaHora,
ZoneId.systemDefault());

    // Obtener todas las zonas horarias disponibles
    Set<String> zonasHorarias = ZoneId.getAvailableZoneIds();

    // Mostrar la fecha y la hora en todas las zonas horarias
    System.out.println("Fecha y hora en todas las zonas horarias:");
    for (String zonaHoraria : zonasHorarias) {
                                  ZonedDateTime     zonedDateTimeEnZona     =
zonedDateTime.withZoneSameInstant(ZoneId.of(zonaHoraria));
      System.out
                                        .println(zonaHoraria    +    ":    "    +
zonedDateTimeEnZona.format(DateTimeFormatter.ofPattern("yyyy-MM-dd
HH:mm:ss")));
    }

    scanner.close();
  }
}
```

17. Crea un método que reciba por parámetro la fecha de nacimiento de una persona como LocalDate y que calcule su edad en años.

RESOLUCIÓN:

```
import java.time.LocalDate;
import java.time.Period;

public class App {
  public static void main(String[] args) {
    // Ejemplo de uso del método, dejamos una fecha pre cargada
    LocalDate fechaNacimiento = LocalDate.of(1986, 5, 15);
    int edad = calcularEdadEnAños(fechaNacimiento);
```

Material descargable

egg

Soluciones Actividades

```
    System.out.println("La edad es: " + edad);
  }

  // Método para calcular la edad en años a partir de la fecha de nacimiento
  public static int calcularEdadEnAños(LocalDate fechaNacimiento) {
    // Obtener la fecha actual
    LocalDate fechaActual = LocalDate.now();

    // Calcular el período entre la fecha de nacimiento y la fecha actual
    Period periodo = Period.between(fechaNacimiento, fechaActual);

    // Extraer el componente de años del período
    int años = periodo.getYears();

    return años;
  }
}
```

18. Escribe un programa que tome dos fechas en formato de texto, las convierta a LocalDate y use ChronoUnit para calcular la cantidad de años, meses y días entre las dos fechas.

RESOLUCIÓN:

```
import java.time.LocalDate;
import java.time.temporal.ChronoUnit;
import java.util.Scanner;

public class App {
  public static void main(String[] args) {
    Scanner scanner = new Scanner(System.in);

    // Solicitar al usuario que ingrese las dos fechas en formato de texto
    System.out.println("Ingrese la primera fecha (yyyy-MM-dd):");
    String fechaTexto1 = scanner.nextLine();

    System.out.println("Ingrese la segunda fecha (yyyy-MM-dd):");
    String fechaTexto2 = scanner.nextLine();

    // Convertir las fechas de texto a objetos LocalDate
    LocalDate fecha1 = LocalDate.parse(fechaTexto1);
    LocalDate fecha2 = LocalDate.parse(fechaTexto2);

    // Calcular la cantidad de años, meses y días entre las dos fechas
    long años = ChronoUnit.YEARS.between(fecha1, fecha2);
    long meses = ChronoUnit.MONTHS.between(fecha1, fecha2);
    long días = ChronoUnit.DAYS.between(fecha1, fecha2);
```

Material descargable

egg

Soluciones Actividades

```
    // Mostrar el resultado
     System.out.println("La cantidad de años entre las dos fechas es: " +
años);
     System.out.println("La cantidad de meses entre las dos fechas es: " +
meses);
     System.out.println("La cantidad de días entre las dos fechas es: " +
días);

    scanner.close();
  }
}
```