Informe sobre el libro

las riquezas de las naciones adam smith

App cap 7

jhoan

8 julio 2025

Capítulo 7: Del precio natural y precio de mercado de las mercancías

desarrollo de tu app de simulación de crisis de oferta, usando el escenario de la sequía en el cinturón maicero de EE.UU. (2023) y la API de USDA Quick Stats.

Objetivo:

Simular el impacto de una crisis de oferta (sequía) en el mercado del maíz, mostrando datos reales y simulando efectos económicos (precios, producción, etc.).

1. Estructura General

Backend (Go):

Lógica de simulación.

Conexión y consulta a la API de USDA Quick Stats.

Endpoints para exponer datos y resultados de la simulación.

Frontend (HTML/JS):

Visualización de datos reales y simulados.

Interfaz para ajustar parámetros de la simulación (por ejemplo, severidad de la sequía).

Vamos a crear el endpoint básico para consultar la API de USDA Quick Stats.

Ahora voy a crear la estructura para integrar la API de USDA Quick Stats. Empezaré creando:

1. Modelos para los datos de la API
1. Handlers para manejar las consultas
1. Servicios para la lógica de negocio

1. Actualizar el main.go para incluir los nuevos endpoints

2. Modelos para los datos de la API

```go
package models

// USDAQuickStatsResponse representa la respuesta de la API de USDA Quick Stats
type USDAQuickStatsResponse struct {
    Data []USDADataPoint `json:"data"`
}

// USDADataPoint representa un punto de datos individual de la API
type USDADataPoint struct {
    Year                interface{} `json:"year"` // Puede ser string o number
    StateName           string      `json:"state_name"`
    CommodityDesc       string      `json:"commodity_desc"`
    StatisticcatDesc    string      `json:"statisticcat_desc"`
    Value               string      `json:"Value"`
    UnitDesc            string      `json:"unit_desc"`
    ReferencePeriodDesc string `json:"reference_period_desc"`
    SourceDesc          string      `json:"source_desc"`
    AggLevelDesc        string      `json:"agg_level_desc"`
    UtilPracticeDesc    string      `json:"util_practice_desc"`
    ShortDesc           string      `json:"short_desc"`
}


// CornProductionData representa datos específicos de producción de maíz
type CornProductionData struct {
    Year       string  `json:"year"`
    State      string  `json:"state"`
    Production float64 `json:"production"`
    Unit       string  `json:"unit"`
    Area       float64 `json:"area,omitempty"`
    AreaUnit   string  `json:"area_unit,omitempty"`
}


// DroughtSimulationParams parámetros para la simulación de sequía
type DroughtSimulationParams struct {
```

```go
    Year             string   `json:"year"`
    States           []string `json:"states"`
    DroughtSeverity  float64  `json:"drought_severity"` // 0.0 a 1.0 (0% a 100% de reducción)
    AffectedArea     float64  `json:"affected_area"`    // Porcentaje del área afectada
}

// DroughtSimulationResult resultado de la simulación
type DroughtSimulationResult struct {
    OriginalProduction  float64 `json:"original_production"`
    SimulatedProduction float64 `json:"simulated_production"`
    ProductionLoss      float64 `json:"production_loss"`
    PriceIncrease       float64 `json:"price_increase"`
    EconomicImpact      float64 `json:"economic_impact"`
    States              []CornProductionData `json:"states"`
}
```

1. Ahora voy a crear el servicio para manejar las consultas a la API de USDA

```go
package services

import (
    "encoding/json"
    "fmt"
    "io"
    "net/http"
    "net/url"
    "strconv"
    "strings"

"github.com/jhoan28310576/cap7-8-9_las_riquesas_de_las_naciones/internal/models"
)

const (
    USDA_API_BASE_URL = "https://quickstats.nass.usda.gov/api/api_GET/"
    USDA_API_KEY      = "1F325726-42E7-3E08-8E7F-C7ED7047890A"
)

type USDAService struct {
    apiKey string
    client *http.Client
}

func NewUSDAService() *USDAService {
    return &USDAService{
        apiKey: USDA_API_KEY,
        client: &http.Client{},
    }
}

// GetCornProduction obtiene datos de producción de maíz para estados específicos
func (s *USDAService) GetCornProduction(year string, states []string) ([]models.CornProductionData, error) {
    var allData []models.CornProductionData
    for _, state := range states {
        // Consultar producción
        productionData, err := s.queryUSDA("CORN", year, state, "PRODUCTION")
        if err != nil {
            return nil, fmt.Errorf("error querying production for %s: %v", state, err)
        }

        // Consultar área cosechada
        areaData, err := s.queryUSDA("CORN", year, state, "AREA HARVESTED")
        if err != nil {
            return nil, fmt.Errorf("error querying area for %s: %v", state, err)
        }

        // Procesar datos de producción
        for _, data := range productionData.Data {
            // Filtrar solo datos de nivel estatal y de grano (no silage)
            if data.AggLevelDesc != "STATE" || data.UtilPracticeDesc != "GRAIN" {
                continue
            }


            production, err :=
parseNumericValue(data.Value)
            if err != nil {
                continue // Skip invalid data
            }


            // Convertir year a string
            yearStr := ""
            switch v := data.Year.(type) {
            case string:
                yearStr = v
            case float64:
                yearStr = fmt.Sprintf("%.0f", v)
            case int:
                yearStr = fmt.Sprintf("%d", v)
            default:
                yearStr = fmt.Sprintf("%v", v)
            }
            cornData := models.CornProductionData{
                Year:       yearStr,
                State:      data.StateName,
                Production: production,
                Unit:       data.UnitDesc,
            }

            // Buscar área correspondiente
            for _, area := range areaData.Data {
                // Filtrar solo datos de nivel estatal y de grano
                if area.AggLevelDesc != "STATE" || area.UtilPracticeDesc != "GRAIN" {
                    continue
                }

                // Convertir year del área a string para comparar
                areaYearStr := ""
                switch v := area.Year.(type) {
                case string:
                    areaYearStr = v
                case float64:
                    areaYearStr = fmt.Sprintf("%.0f", v)
                case int:
                    areaYearStr = fmt.Sprintf("%d", v)
                default:
                    areaYearStr = fmt.Sprintf("%v", v)
                }

                if areaYearStr == yearStr && area.StateName == data.StateName {
                    if areaValue, err := parseNumericValue(area.Value); err == nil {
                        cornData.Area = areaValue
                        cornData.AreaUnit = area.UnitDesc
                    }
                    break
                }
            }

            allData = append(allData, cornData)
```

```
        }
    }

    return allData, nil
}

// queryUSDA hace una consulta a la API de USDA Quick Stats
func (s *USDAService) queryUSDA(commodity, year, state, statistic string) (*models.USDAQuickStatsResponse, error) {
    baseURL, err := url.Parse(USDA_API_BASE_URL)
    if err != nil {
        return nil, err
    }

    params := url.Values{}
    params.Add("key", s.apiKey)
    params.Add("commodity_desc", commodity)
    params.Add("year", year)
    params.Add("state_alpha", state)
    params.Add("statisticcat_desc", statistic)
    params.Add("format", "JSON")
    baseURL.RawQuery = params.Encode()

    resp, err := s.client.Get(baseURL.String())
    if err != nil {
        return nil, fmt.Errorf("error making request: %v", err)
    }
    defer resp.Body.Close()

    if resp.StatusCode != http.StatusOK {
        body, _ := io.ReadAll(resp.Body)
        return nil, fmt.Errorf("API returned status %d: %s", resp.StatusCode, string(body))
    }

    body, err := io.ReadAll(resp.Body)
    if err != nil {
        return nil, fmt.Errorf("error reading response: %v", err)
    }
```

```go
    var response models.USDAQuickStatsResponse
    if err := json.Unmarshal(body, &response); err != nil {
        return nil, fmt.Errorf("error parsing JSON: %v", err)
    }

    return &response, nil
}

// parseNumericValue convierte el valor string de la API a float64
func parseNumericValue(valueStr string) (float64, error) {
    // Remover comas y espacios
    cleanValue := strings.ReplaceAll(valueStr, ",", "")
    cleanValue = strings.TrimSpace(cleanValue)

    return strconv.ParseFloat(cleanValue, 64)
}

// SimulateDrought simula el impacto de una sequía en la producción de maíz
```

```
func (s *USDAService) SimulateDrought(params models.DroughtSimulationParams) (*models.DroughtSimulationResult, error) {
    // Obtener datos reales
    realData, err := s.GetCornProduction(params.Year, params.States)
    if err != nil {
        return nil, err
    }

    var originalProduction, simulatedProduction float64
    var simulatedStates []models.CornProductionData

    for _, data := range realData {
        originalProduction += data.Production

        // Calcular producción simulada
        affectedProduction := data.Production * params.AffectedArea * params.DroughtSeverity
        unaffectedProduction := data.Production * (1 - params.AffectedArea)
        newProduction := unaffectedProduction + (affectedProduction * (1 - params.DroughtSeverity))
        simulatedData := data
        simulatedData.Production = newProduction
        simulatedStates = append(simulatedStates,
simulatedData)
        simulatedProduction += newProduction
    }

    productionLoss := originalProduction -
simulatedProduction
    priceIncrease := (productionLoss / originalProduction)
* 0.5 // Elasticidad de precio simplificada
    economicImpact := productionLoss * 4.5
// Valor aproximado por bushel

    return &models.DroughtSimulationResult{
        OriginalProduction:  originalProduction,
        SimulatedProduction: simulatedProduction,
        ProductionLoss:      productionLoss,
        PriceIncrease:       priceIncrease,
        EconomicImpact:      economicImpact,
        States:              simulatedStates,
    }, nil
}
```

Ahora voy a crear los handlers para manejar las peticiones HTTP:

```
package handlers

import (
    "net/http"
    "strings"

"github.com/jhoan28310576/cap7-8-9_las_riquesas_de_las_naciones/internal/models"
"github.com/jhoan28310576/cap7-8-9_las_riquesas_de_las_naciones/internal/services"

    "github.com/gin-gonic/gin"
)

type USDAHandlers struct {
    usdaService *services.USDAService
}

func NewUSDAHandlers() *USDAHandlers {
    return &USDAHandlers{
        usdaService: services.NewUSDAService(),
    }
}

// GetCornProduction obtiene datos de producción de maíz
func (h *USDAHandlers) GetCornProduction(c *gin.Context) {
    year := c.Query("year")
    if year == "" {
        year = "2023" // Default year
    }

    statesParam := c.Query("states")
    if statesParam == "" {
        // Estados del cinturón maicero por defecto
        statesParam = "IA,IL,NE,MN,IN,OH,WI,SD,MO,KS"
    }
```

```go
    states := []string{"IA", "IL", "NE", "MN", "IN", "OH", "WI", "SD", "MO", "KS"}
    if statesParam != "" {
        // Parse comma-separated states
        states = []string{}
        for _, state := range strings.Split(statesParam, ",") {
            states = append(states, strings.TrimSpace(state))
        }
    }

    data, err := h.usdaService.GetCornProduction(year, states)
    if err != nil {
        c.JSON(http.StatusInternalServerError, gin.H{
            "error": "Error obteniendo datos de producción: " + err.Error(),
        })
        return
    }

    c.JSON(http.StatusOK, gin.H{
        "year":   year,
        "states": states,
        "data":   data,
    })
}

// SimulateDrought simula el impacto de una sequía
func (h *USDAHandlers) SimulateDrought(c *gin.Context) {
    var params models.DroughtSimulationParams

    if err := c.ShouldBindJSON(&params); err != nil {
        c.JSON(http.StatusBadRequest, gin.H{
            "error": "Datos de entrada inválidos: " +
err.Error(),
        })
        return
    }

    // Validar parámetros
    if params.Year == "" {
        params.Year = "2023"
    }
    if len(params.States) == 0 {
        params.States = []string{"IA", "IL", "NE", "MN", "IN", "OH", "WI", "SD", "MO", "KS"}
    }
    if params.DroughtSeverity < 0 || params.DroughtSeverity > 1 {
        c.JSON(http.StatusBadRequest, gin.H{
            "error": "La severidad de la sequía debe estar entre 0 y 1",
        })
        return
    }
    if params.AffectedArea < 0 || params.AffectedArea > 1 {
        c.JSON(http.StatusBadRequest, gin.H{
            "error": "El área afectada debe estar entre 0 y 1",
        })
        return
    }

    result, err := h.usdaService.SimulateDrought(params)
    if err != nil {
        c.JSON(http.StatusInternalServerError, gin.H{
            "error": "Error en la simulación: " + err.Error(),
        })
        return
    }

    c.JSON(http.StatusOK, gin.H{
        "simulation_params": params,
        "result":            result,
    })
}

// GetDroughtSimulationForm muestra un formulario para la
simulación
func (h *USDAHandlers) GetDroughtSimulationForm(c *gin.Context)
{
    c.HTML(http.StatusOK, "drought_simulation.html", gin.H{
        "title": "Simulación de Crisis de Oferta - Sequía en el
Cinturón Maicero",
    })
}
```

```
// GetCornProductionByState obtiene datos de un estado específico
func (h *USDAHandlers) GetCornProductionByState(c *gin.Context) {
    state := c.Param("state")
    year := c.Query("year")
    if year == "" {
        year = "2023"
    }

    data, err := h.usdaService.GetCornProduction(year, []string{state})
    if err != nil {
        c.JSON(http.StatusInternalServerError, gin.H{
            "error": "Error obteniendo datos: " + err.Error(),
        })
        return
    }

    if len(data) == 0 {
        c.JSON(http.StatusNotFound, gin.H{
            "error": "No se encontraron datos para el estado " +
state,
        })
        return
    }

    c.JSON(http.StatusOK, gin.H{
        "state": state,
        "year":  year,
        "data":  data[0],
    })
}
```

5)Actualizar el main.go para incluir los nuevos endpoints

```
// Inicializar handlers
    usdaHandlers := handlers.NewUSDAHandlers()

// API endpoints para USDA
    api := r.Group("/api")
    {
        api.GET("/corn-production", usdaHandlers.GetCornProduction)
        api.GET("/corn-production/:state", usdaHandlers.GetCornProductionByState)
        api.POST("/drought-simulation", usdaHandlers.SimulateDrought)
    }

    // Página de simulación
    r.GET("/simulation", usdaHandlers.GetDroughtSimulationForm)
```

Ahora voy a crear la página HTML para la simulación de sequía:

```
<!DOCTYPE html>
<html lang="es">
  <head>
    <meta charset="UTF-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <meta http-equiv="X-UA-Compatible" content="ie=edge" />
```

```html
    <title>Simulación de Crisis de Oferta</title>
    <link rel="stylesheet" href="/assets/css/style.css" />
    <link rel="stylesheet" href="/assets/css/nav.css" />
  </head>
  <body>
    <nav>
      <div class="logo">
        <img src="assets/logo/Logo64x64.png" alt="logo" />
        <h1>LOGO</h1>
      </div>
      <ul>
        <li>
          <a href="#">Capítulo 7:Del precio natural y precio de mercado de las mercancías</a>
        </li>
        <li>
          <a href="#">Capítulo 8: De los salarios del trabajoervices</a>
        </li>
        <li>
          <a href="#">Capítulo 9: De los beneficios del capital</a>
        </li>
        <li>
          <a href="#">Siguientes capítulos</a>
        </li>
      </ul>
      <div class="hamburger">
        <span class="line"></span>
        <span class="line"></span>
        <span class="line"></span>
      </div>
    </nav>
    <div class="menubar">
      <ul>
        <li>
          <a href="#">capitulo 7: Del precio natural y precio de mercado de las mercancías</a>
        </li>
        <li>
          <a href="#">capitulo 8: De los salarios del trabajo</a>
        </li>
        <li>
          <a href="#">capitulo 9: De los beneficios del capital</a>
        </li>
        <li>
          <a href="#">Siguientes capítulos</a>
        </li>
      </ul>
    </div>

    <br />

    <div class="container">
      <div class="header">
        <h1>🌾 Simulación de Crisis de Oferta</h1>
        <p>Sequía en el Cinturón Maicero de EE.UU. (2023) - Capítulo 7: Las Riquezas de las Naciones</p>
      </div>

      <div class="content">
        <div class="simulation-form">
```

```
          <h2>📊 Parámetros de la Simulación</h2>

          <div class="form-group">
            <label for="year">Año de Referencia:</label>
            <select id="year">
              <option value="2023">2023</option>
              <option value="2022">2022</option>
              <option value="2021">2021</option>
              <option value="2020">2020</option>
            </select>
          </div>

          <div class="form-group">
            <label for="states">Estados del Cinturón Maicero:</label>
            <select id="states" multiple>
              <option value="IA" selected>Iowa (IA)</option>
              <option value="IL" selected>Illinois (IL)</option>
              <option value="NE" selected>Nebraska (NE)</option>
              <option value="MN" selected>Minnesota (MN)</option>
```

```html
              <option value="IN" selected>Indiana (IN)</option>
              <option value="OH" selected>Ohio (OH)</option>
              <option value="WI" selected>Wisconsin (WI)</option>
              <option value="SD" selected>South Dakota (SD)</option>
              <option value="MO" selected>Missouri (MO)</option>
              <option value="KS" selected>Kansas (KS)</option>
            </select>
          </div>

          <div class="form-group">
            <label>Severidad de la Sequía (0% - 100%):</label>
            <div class="slider-container">
              <input type="range" id="droughtSeverity" class="slider" min="0" max="100" value="30" />
              <span id="droughtSeverityValue" class="slider-value">30%</span>
            </div>
          </div>

          <div class="form-group">
```

```
            <label>Área Afectada (0% - 100%):</label>
            <div class="slider-container">
              <input type="range" id="affectedArea"
class="slider" min="0" max="100" value="50" />
              <span id="affectedAreaValue"
class="slider-value">50%</span>
            </div>
          </div>

          <button onclick="runSimulation()">🚀 Ejecutar
Simulación</button>
        </div>

        <div id="loading" class="loading" style="display:
none;">
          <h3>🔄 Consultando datos de USDA...</h3>
          <p>Obteniendo datos reales de producción de
maíz...</p>
        </div>

        <div id="error" class="error" style="display:
none;"></div>
```

```html
        <div id="results" class="results">
          <div class="result-card">
            <h3>📈 Impacto Económico de la Sequía</h3>
            <div class="metric">
              <span class="metric-label">Producción Original:</span>
              <span id="originalProduction" class="metric-value">-</span>
            </div>
            <div class="metric">
              <span class="metric-label">Producción Simulada:</span>
              <span id="simulatedProduction" class="metric-value">-</span>
            </div>
            <div class="metric">
              <span class="metric-label">Pérdida de Producción:</span>
              <span id="productionLoss" class="metric-value">-</span>
            </div>
            <div class="metric">
```

```
              <span class="metric-label">Incremento en Precios:</span>
              <span id="priceIncrease" class="metric-value">-</span>
            </div>
            <div class="metric">
              <span class="metric-label">Impacto Económico:</span>
              <span id="economicImpact" class="metric-value">-</span>
            </div>
          </div>

          <div class="result-card">
            <h3>🗺️ Producción por Estado</h3>
            <div id="statesGrid" class="states-grid"></div>
          </div>
        </div>
      </div>
    </div>

    <script src="/assets/js/script.js"></script>
    <script src="/assets/js/nav.js"></script>
  </body>
</html>
```

resolviendo error USDADataPoint.data.year of type string

(al ejecutar el simulador me dice Error: Error en la simulación: error querying production for IA: error parsing JSON: json: cannot unmarshal number into Go struct field USDADataPoint.data.year of type string , vamos a detectar el problema y por q pasa)

El error indica que la API de USDA está devolviendo el campo year como un número, pero nuestro modelo lo está esperando como un string. Vamos a investigar la estructura real de la respuesta de la API y corregir el modelo

Vamos a crear un pequeño programa Go para probar la API y ver la estructura exacta de la respuesta:

creando test_api.go

```
package main

import (
    "encoding/json"
    "fmt"
    "io"
    "net/http"
    "net/url"
)
```

```go
func main() {
    baseURL, _ := url.Parse("https://quickstats.nass.usda.gov/api/api_GET/")

    params := url.Values{}
    params.Add("key", "1F325726-42E7-3E08-8E7F-C7ED7047890A")
    params.Add("commodity_desc", "CORN")
    params.Add("year", "2023")
    params.Add("state_alpha", "IA")
    params.Add("statisticcat_desc", "PRODUCTION")
    params.Add("format", "JSON")

    baseURL.RawQuery = params.Encode()

    resp, err := http.Get(baseURL.String())
    if err != nil {
        fmt.Printf("Error: %v\n", err)
        return
    }
    defer resp.Body.Close()
    body, err := io.ReadAll(resp.Body)
    if err != nil {
        fmt.Printf("Error reading response: %v\n", err)
        return
    }

    fmt.Printf("Status: %d\n", resp.StatusCode)
    fmt.Printf("Response:\n%s\n", string(body))

    // Intentar parsear como JSON genérico para ver la estructura
    var rawData map[string]interface{}
    if err := json.Unmarshal(body, &rawData); err != nil {
        fmt.Printf("Error parsing JSON: %v\n", err)
        return
    }

    // Imprimir la estructura
    prettyJSON, _ := json.MarshalIndent(rawData, "", "  ")
    fmt.Printf("Parsed structure:\n%s\n", string(prettyJSON))
}
```

ejecuto el test_api respuesta completa

```
"statisticcat_desc": "PRODUCTION",
"unit_desc": "BU",
"util_practice_desc": "GRAIN",
"watershed_code": "00000000",
"watershed_desc": "",
"week_ending": "",
"year": 2023,
"zip_5": ""
},
{
"CV (%)": "",
"Value": "2,520,000,000",
"agg_level_desc": "STATE",
"asd_code": "",
"asd_desc": "",
"begin_code": "00",
"class_desc": "ALL CLASSES",
"commodity_desc": "CORN",
"congr_district_code": "",
"country_code": "9000",
"country_name": "UNITED STATES",
"county_ansi": "",
"county_code": "",
"county_name": "",
"domain_desc": "TOTAL",
"domaincat_desc": "NOT SPECIFIED",
"end_code": "00",
"freq_desc": "ANNUAL",
"group_desc": "FIELD CROPS",
"load_time": "2023-09-12 12:00:00.000",
"location_desc": "IOWA",
"prodn_practice_desc": "ALL PRODUCTION PRACTICES",
"reference_period_desc": "YEAR - SEP FORECAST",
"region_desc": "",
"sector_desc": "CROPS",
"short_desc": "CORN, GRAIN - PRODUCTION, MEASURED IN BU",
"source_desc": "SURVEY",
"state_alpha": "IA",
"state_ansi": "19",
"state_fips_code": "19",
"state_name": "IOWA",
"statisticcat_desc": "PRODUCTION",
"unit_desc": "BU",
"util_practice_desc": "GRAIN",
"watershed_code": "00000000",
"watershed_desc": "",
"week_ending": "",
"year": 2023,
"zip_5": ""
},
{
"CV (%)": "7.6",
"Value": "51,000",
"agg_level_desc": "COUNTY",
"asd_code": "10",
"asd_desc": "NORTHWEST",
"begin_code": "00",
"class_desc": "ALL CLASSES",
"commodity_desc": "CORN",
"congr_district_code": "",
"country_code": "9000",
"country_name": "UNITED STATES",
"county_ansi": "035",
"county_code": "035",
"county_name": "CHEROKEE",
"domain_desc": "TOTAL",
"domaincat_desc": "NOT SPECIFIED",
"end_code": "00",
"freq_desc": "ANNUAL",
"group_desc": "FIELD CROPS",
"load_time": "2025-05-06 15:00:00.000",
"location_desc": "IOWA, NORTHWEST, CHEROKEE",
"prodn_practice_desc": "ALL PRODUCTION PRACTICES",
"reference_period_desc": "YEAR",
"region_desc": "",
"sector_desc": "CROPS",
"short_desc": "CORN, SILAGE - PRODUCTION, MEASURED IN TONS",
"source_desc": "SURVEY",
"state_alpha": "IA",
"state_ansi": "19",
"state_fips_code": "19",
"state_name": "IOWA",
"statisticcat_desc": "PRODUCTION",

"unit_desc": "TONS",

"util_practice_desc": "SILAGE",

"watershed_code": "00000000",

"watershed_desc": "",

"week_ending": "",

"year": 2023,

"zip_5": ""

},

{

"CV (%)": "8.5",

"Value": "85,000",

"agg_level_desc": "COUNTY",

"asd_code": "10",

"asd_desc": "NORTHWEST",

"begin_code": "00",

"class_desc": "ALL CLASSES",

"commodity_desc": "CORN",

"congr_district_code": "",

"country_code": "9000",

"country_name": "UNITED STATES",
"county_ansi": "041",

"county_code": "041",

"county_name": "CLAY",

"domain_desc": "TOTAL",

"domaincat_desc": "NOT SPECIFIED",

"end_code": "00",

"freq_desc": "ANNUAL",

"group_desc": "FIELD CROPS",

"load_time": "2025-05-06 15:00:00.000",

"location_desc": "IOWA, NORTHWEST, CLAY",

"prodn_practice_desc": "ALL PRODUCTION PRACTICES",

"reference_period_desc": "YEAR",

"region_desc": "",

"sector_desc": "CROPS",

"short_desc": "CORN, SILAGE - PRODUCTION, MEASURED IN TONS",

"source_desc": "SURVEY",

"state_alpha": "IA",

"state_ansi": "19",

"state_fips_code": "19",

"state_name": "IOWA",

"statisticcat_desc": "PRODUCTION",
"unit_desc": "TONS",
"util_practice_desc": "SILAGE",
"watershed_code": "00000000",
"watershed_desc": "",
"week_ending": "",
"year": 2023,
"zip_5": ""
},
{
"CV (%)": "5.9",
"Value": "400,000",
"agg_level_desc": "COUNTY",
"asd_code": "10",
"asd_desc": "NORTHWEST",
"begin_code": "00",
"class_desc": "ALL CLASSES",
"commodity_desc": "CORN",
"congr_district_code": "",
"country_code": "9000",
"country_name": "UNITED STATES",
"county_ansi": "119",
"county_code": "119",
"county_name": "LYON",
"domain_desc": "TOTAL",
"domaincat_desc": "NOT SPECIFIED",
"end_code": "00",
"freq_desc": "ANNUAL",
"group_desc": "FIELD CROPS",
"load_time": "2025-05-06 15:00:00.000",
"location_desc": "IOWA, NORTHWEST, LYON",
"prodn_practice_desc": "ALL PRODUCTION PRACTICES",
"reference_period_desc": "YEAR",
"region_desc": "",
"sector_desc": "CROPS",
"short_desc": "CORN, SILAGE - PRODUCTION, MEASURED IN TONS",
"source_desc": "SURVEY",
"state_alpha": "IA",
"state_ansi": "19",
"state_fips_code": "19",
"state_name": "IOWA",
"statisticcat_desc": "PRODUCTION",
"unit_desc": "TONS",
"util_practice_desc": "SILAGE",

"watershed_code": "00000000",

"watershed_desc": "",

"week_ending": "",

"year": 2023,

"zip_5": ""

},

{

"CV (%)": "6.4",

"Value": "128,000",

"agg_level_desc": "COUNTY",

"asd_code": "10",

"asd_desc": "NORTHWEST",

"begin_code": "00",

"class_desc": "ALL CLASSES",

"commodity_desc": "CORN",

"congr_district_code": "",

"country_code": "9000",

"country_name": "UNITED STATES",

"county_ansi": "141",

"county_code": "141",
"county_name": "O BRIEN",

"domain_desc": "TOTAL",

"domaincat_desc": "NOT SPECIFIED",

"end_code": "00",

"freq_desc": "ANNUAL",

"group_desc": "FIELD CROPS",

"load_time": "2025-05-06 15:00:00.000",

"location_desc": "IOWA, NORTHWEST, O BRIEN",

"prodn_practice_desc": "ALL PRODUCTION PRACTICES",

"reference_period_desc": "YEAR",

"region_desc": "",

"sector_desc": "CROPS",

"short_desc": "CORN, SILAGE - PRODUCTION, MEASURED IN TONS",

"source_desc": "SURVEY",

"state_alpha": "IA",

"state_ansi": "19",

"state_fips_code": "19",

"state_name": "IOWA",

"statisticcat_desc": "PRODUCTION",

"unit_desc": "TONS",

"util_practice_desc": "SILAGE",
"watershed_code": "00000000",
"watershed_desc": "",
"week_ending": "",
"year": 2023,
"zip_5": ""
},
{
"CV (%)": "2.3",
"Value": "723,000",
"agg_level_desc": "COUNTY",
"asd_code": "10",
"asd_desc": "NORTHWEST",
"begin_code": "00",
"class_desc": "ALL CLASSES",
"commodity_desc": "CORN",
"congr_district_code": "",
"country_code": "9000",
"country_name": "UNITED STATES",
"county_ansi": "167",
"county_code": "167",
"county_name": "SIOUX",
"domain_desc": "TOTAL",

"domaincat_desc": "NOT SPECIFIED",

"end_code": "00",

"freq_desc": "ANNUAL",

"group_desc": "FIELD CROPS",

"load_time": "2025-05-06 15:00:00.000",

"location_desc": "IOWA, NORTHWEST, SIOUX",

"prodn_practice_desc": "ALL PRODUCTION PRACTICES",

"reference_period_desc": "YEAR",

"region_desc": "",

"sector_desc": "CROPS",

"short_desc": "CORN, SILAGE - PRODUCTION, MEASURED IN TONS",

"source_desc": "SURVEY",

"state_alpha": "IA",

"state_ansi": "19",

"state_fips_code": "19",

"state_name": "IOWA",

"statisticcat_desc": "PRODUCTION",

"unit_desc": "TONS",

"util_practice_desc": "SILAGE",

"watershed_code": "00000000",
"watershed_desc": "",
"week_ending": "",
"year": 2023,
"zip_5": ""
},
{
"CV (%)": "6.7",
"Value": "226,000",
"agg_level_desc": "COUNTY",
"asd_code": "20",
"asd_desc": "NORTH CENTRAL",
"begin_code": "00",
"class_desc": "ALL CLASSES",
"commodity_desc": "CORN",
"congr_district_code": "",
"country_code": "9000",
"country_name": "UNITED STATES",
"county_ansi": "067",
"county_code": "067",
"county_name": "FLOYD",
"domain_desc": "TOTAL",
"domaincat_desc": "NOT SPECIFIED",

"end_code": "00",

"freq_desc": "ANNUAL",

"group_desc": "FIELD CROPS",

"load_time": "2025-05-06 15:00:00.000",

"location_desc": "IOWA, NORTH CENTRAL, FLOYD",

"prodn_practice_desc": "ALL PRODUCTION PRACTICES",

"reference_period_desc": "YEAR",

"region_desc": "",

"sector_desc": "CROPS",

"short_desc": "CORN, SILAGE - PRODUCTION, MEASURED IN TONS",

"source_desc": "SURVEY",

"state_alpha": "IA",

"state_ansi": "19",

"state_fips_code": "19",

"state_name": "IOWA",

"statisticcat_desc": "PRODUCTION",

"unit_desc": "TONS",

"util_practice_desc": "SILAGE",

"watershed_code": "00000000",

"watershed_desc": "",
"week_ending": "",

"year": 2023,

"zip_5": ""

},

{

"CV (%)": "8.3",

"Value": "175,000",

"agg_level_desc": "COUNTY",

"asd_code": "30",

"asd_desc": "NORTHEAST",

"begin_code": "00",

"class_desc": "ALL CLASSES",

"commodity_desc": "CORN",

"congr_district_code": "",

"country_code": "9000",

"country_name": "UNITED STATES",

"county_ansi": "005",

"county_code": "005",

"county_name": "ALLAMAKEE",

"domain_desc": "TOTAL",

"domaincat_desc": "NOT SPECIFIED",
"end_code": "00",
"freq_desc": "ANNUAL",
"group_desc": "FIELD CROPS",
"load_time": "2025-05-06 15:00:00.000",
"location_desc": "IOWA, NORTHEAST, ALLAMAKEE",
"prodn_practice_desc": "ALL PRODUCTION PRACTICES",
"reference_period_desc": "YEAR",
"region_desc": "",
"sector_desc": "CROPS",
"short_desc": "CORN, SILAGE - PRODUCTION, MEASURED IN TONS",
"source_desc": "SURVEY",
"state_alpha": "IA",
"state_ansi": "19",
"state_fips_code": "19",
"state_name": "IOWA",
"statisticcat_desc": "PRODUCTION",
"unit_desc": "TONS",
"util_practice_desc": "SILAGE",
"watershed_code": "00000000",
"watershed_desc": "",
"week_ending": "",
"year": 2023,

"zip_5": ""

},

{

"CV (%)": "4.5",

"Value": "55,000",

"agg_level_desc": "COUNTY",

"asd_code": "30",

"asd_desc": "NORTHEAST",

"begin_code": "00",

"class_desc": "ALL CLASSES",

"commodity_desc": "CORN",

"congr_district_code": "",

"country_code": "9000",

"country_name": "UNITED STATES",

"county_ansi": "017",

"county_code": "017",

"county_name": "BREMER",

"domain_desc": "TOTAL",

"domaincat_desc": "NOT SPECIFIED",

"end_code": "00",
"freq_desc": "ANNUAL",
"group_desc": "FIELD CROPS",
"load_time": "2025-05-06 15:00:00.000",
"location_desc": "IOWA, NORTHEAST, BREMER",
"prodn_practice_desc": "ALL PRODUCTION PRACTICES",
"reference_period_desc": "YEAR",
"region_desc": "",
"sector_desc": "CROPS",
"short_desc": "CORN, SILAGE - PRODUCTION, MEASURED IN TONS",
"source_desc": "SURVEY",
"state_alpha": "IA",
"state_ansi": "19",
"state_fips_code": "19",
"state_name": "IOWA",
"statisticcat_desc": "PRODUCTION",
"unit_desc": "TONS",
"util_practice_desc": "SILAGE",
"watershed_code": "00000000",
"watershed_desc": "",
"week_ending": "",
"year": 2023,
"zip_5": ""

},

{

"CV (%)": "12.0",

"Value": "129,000",

"agg_level_desc": "COUNTY",

"asd_code": "30",

"asd_desc": "NORTHEAST",

"begin_code": "00",

"class_desc": "ALL CLASSES",

"commodity_desc": "CORN",

"congr_district_code": "",

"country_code": "9000",

"country_name": "UNITED STATES",

"county_ansi": "037",

"county_code": "037",

"county_name": "CHICKASAW",

"domain_desc": "TOTAL",

"domaincat_desc": "NOT SPECIFIED",

"end_code": "00",

"freq_desc": "ANNUAL",
"group_desc": "FIELD CROPS",
"load_time": "2025-05-06 15:00:00.000",
"location_desc": "IOWA, NORTHEAST, CHICKASAW",
"prodn_practice_desc": "ALL PRODUCTION PRACTICES",
"reference_period_desc": "YEAR",
"region_desc": "",
"sector_desc": "CROPS",
"short_desc": "CORN, SILAGE - PRODUCTION, MEASURED IN TONS",
"source_desc": "SURVEY",
"state_alpha": "IA",
"state_ansi": "19",
"state_fips_code": "19",
"state_name": "IOWA",
"statisticcat_desc": "PRODUCTION",
"unit_desc": "TONS",
"util_practice_desc": "SILAGE",
"watershed_code": "00000000",
"watershed_desc": "",
"week_ending": "",
"year": 2023,
"zip_5": ""
```

```
},
{
"CV (%)": "5.9",
"Value": "163,000",
"agg_level_desc": "COUNTY",
"asd_code": "30",
"asd_desc": "NORTHEAST",
"begin_code": "00",
"class_desc": "ALL CLASSES",
"commodity_desc": "CORN",
"congr_district_code": "",
"country_code": "9000",
"country_name": "UNITED STATES",
"county_ansi": "055",
"county_code": "055",
"county_name": "DELAWARE",
"domain_desc": "TOTAL",
"domaincat_desc": "NOT SPECIFIED",
"end_code": "00",
"freq_desc": "ANNUAL",
"group_desc": "FIELD CROPS",
"load_time": "2025-05-06 15:00:00.000",

"location_desc": "IOWA, NORTHEAST, DELAWARE",

"prodn_practice_desc": "ALL PRODUCTION PRACTICES",

"reference_period_desc": "YEAR",

"region_desc": "",

"sector_desc": "CROPS",

"short_desc": "CORN, SILAGE - PRODUCTION, MEASURED IN TONS",

"source_desc": "SURVEY",

"state_alpha": "IA",

"state_ansi": "19",

"state_fips_code": "19",

"state_name": "IOWA",

"statisticcat_desc": "PRODUCTION",

"unit_desc": "TONS",

"util_practice_desc": "SILAGE",

"watershed_code": "00000000",

"watershed_desc": "",

"week_ending": "",

"year": 2023,

"zip_5": ""

},
{
"CV (%)": "2.3",
"Value": "258,000",
"agg_level_desc": "COUNTY",
"asd_code": "30",
"asd_desc": "NORTHEAST",
"begin_code": "00",
"class_desc": "ALL CLASSES",
"commodity_desc": "CORN",
"congr_district_code": "",
"country_code": "9000",
"country_name": "UNITED STATES",
"county_ansi": "061",
"county_code": "061",
"county_name": "DUBUQUE",
"domain_desc": "TOTAL",
"domaincat_desc": "NOT SPECIFIED",
"end_code": "00",
"freq_desc": "ANNUAL",
"group_desc": "FIELD CROPS",
"load_time": "2025-05-06 15:00:00.000",
"location_desc": "IOWA, NORTHEAST, DUBUQUE",

"prodn_practice_desc": "ALL PRODUCTION PRACTICES",

"reference_period_desc": "YEAR",

"region_desc": "",

"sector_desc": "CROPS",

"short_desc": "CORN, SILAGE - PRODUCTION, MEASURED IN TONS",

"source_desc": "SURVEY",

"state_alpha": "IA",

"state_ansi": "19",

"state_fips_code": "19",

"state_name": "IOWA",

"statisticcat_desc": "PRODUCTION",

"unit_desc": "TONS",

"util_practice_desc": "SILAGE",

"watershed_code": "00000000",

"watershed_desc": "",

"week_ending": "",

"year": 2023,

"zip_5": ""

},

{
```

```
"CV (%)": "4.3",

"Value": "129,000",

"agg_level_desc": "COUNTY",

"asd_code": "30",

"asd_desc": "NORTHEAST",

"begin_code": "00",

"class_desc": "ALL CLASSES",

"commodity_desc": "CORN",

"congr_district_code": "",

"country_code": "9000",

"country_name": "UNITED STATES",

"county_ansi": "065",

"county_code": "065",

"county_name": "FAYETTE",

"domain_desc": "TOTAL",

"domaincat_desc": "NOT SPECIFIED",

"end_code": "00",

"freq_desc": "ANNUAL",

"group_desc": "FIELD CROPS",

"load_time": "2025-05-06 15:00:00.000",

"location_desc": "IOWA, NORTHEAST, FAYETTE",
"prodn_practice_desc": "ALL PRODUCTION PRACTICES",
"reference_period_desc": "YEAR",
"region_desc": "",
"sector_desc": "CROPS",
"short_desc": "CORN, SILAGE - PRODUCTION, MEASURED IN TONS",
"source_desc": "SURVEY",
"state_alpha": "IA",
"state_ansi": "19",
"state_fips_code": "19",
"state_name": "IOWA",
"statisticcat_desc": "PRODUCTION",
"unit_desc": "TONS",
"util_practice_desc": "SILAGE",
"watershed_code": "00000000",
"watershed_desc": "",
"week_ending": "",
"year": 2023,
"zip_5": ""
},
{
"CV (%)": "5.1",
"Value": "418,000",
"agg_level_desc": "COUNTY",
"asd_code": "30",
"asd_desc": "NORTHEAST",
"begin_code": "00",
"class_desc": "ALL CLASSES",
"commodity_desc": "CORN",
"congr_district_code": "",
"country_code": "9000",
"country_name": "UNITED STATES",
"county_ansi": "191",
"county_code": "191",
"county_name": "WINNESHIEK",
"domain_desc": "TOTAL",
"domaincat_desc": "NOT SPECIFIED",
"end_code": "00",
"freq_desc": "ANNUAL",
"group_desc": "FIELD CROPS",
"load_time": "2025-05-06 15:00:00.000",
"location_desc": "IOWA, NORTHEAST, WINNESHIEK",
"prodn_practice_desc": "ALL PRODUCTION PRACTICES",
```

```json
"reference_period_desc": "YEAR",

"region_desc": "",

"sector_desc": "CROPS",

"short_desc": "CORN, SILAGE - PRODUCTION, MEASURED IN TONS",

"source_desc": "SURVEY",

"state_alpha": "IA",

"state_ansi": "19",

"state_fips_code": "19",

"state_name": "IOWA",

"statisticcat_desc": "PRODUCTION",

"unit_desc": "TONS",

"util_practice_desc": "SILAGE",

"watershed_code": "00000000",

"watershed_desc": "",

"week_ending": "",

"year": 2023,

"zip_5": ""

},

{

"CV (%)": "11.2",

"Value": "104,000",
"agg_level_desc": "COUNTY",
"asd_code": "40",
"asd_desc": "WEST CENTRAL",
"begin_code": "00",
"class_desc": "ALL CLASSES",
"commodity_desc": "CORN",
"congr_district_code": "",
"country_code": "9000",
"country_name": "UNITED STATES",
"county_ansi": "193",
"county_code": "193",
"county_name": "WOODBURY",
"domain_desc": "TOTAL",
"domaincat_desc": "NOT SPECIFIED",
"end_code": "00",
"freq_desc": "ANNUAL",
"group_desc": "FIELD CROPS",
"load_time": "2025-05-06 15:00:00.000",
"location_desc": "IOWA, WEST CENTRAL, WOODBURY",
"prodn_practice_desc": "ALL PRODUCTION PRACTICES",
"reference_period_desc": "YEAR",
"region_desc": "",
"sector_desc": "CROPS",
"short_desc": "CORN, SILAGE - PRODUCTION, MEASURED IN TONS",
"source_desc": "SURVEY",
"state_alpha": "IA",
"state_ansi": "19",
"state_fips_code": "19",
"state_name": "IOWA",
"statisticcat_desc": "PRODUCTION",
"unit_desc": "TONS",
"util_practice_desc": "SILAGE",
"watershed_code": "00000000",
"watershed_desc": "",
"week_ending": "",
"year": 2023,
"zip_5": ""
},
{
"CV (%)": "4.0",
"Value": "318,000",
"agg_level_desc": "COUNTY",
"asd_code": "50",
"asd_desc": "CENTRAL",
"begin_code": "00",
"class_desc": "ALL CLASSES",
"commodity_desc": "CORN",
"congr_district_code": "",
"country_code": "9000",
"country_name": "UNITED STATES",
"county_ansi": "127",
"county_code": "127",
"county_name": "MARSHALL",
"domain_desc": "TOTAL",
"domaincat_desc": "NOT SPECIFIED",
"end_code": "00",
"freq_desc": "ANNUAL",
"group_desc": "FIELD CROPS",
"load_time": "2025-05-06 15:00:00.000",
"location_desc": "IOWA, CENTRAL, MARSHALL",
"prodn_practice_desc": "ALL PRODUCTION PRACTICES",
"reference_period_desc": "YEAR",
"region_desc": "",
"sector_desc": "CROPS",

"short_desc": "CORN, SILAGE - PRODUCTION, MEASURED IN TONS",

"source_desc": "SURVEY",

"state_alpha": "IA",

"state_ansi": "19",

"state_fips_code": "19",

"state_name": "IOWA",

"statisticcat_desc": "PRODUCTION",

"unit_desc": "TONS",

"util_practice_desc": "SILAGE",

"watershed_code": "00000000",

"watershed_desc": "",

"week_ending": "",

"year": 2023,

"zip_5": ""

},

{

"CV (%)": "7.7",

"Value": "25,700",

"agg_level_desc": "COUNTY",

"asd_code": "60",
"asd_desc": "EAST CENTRAL",
"begin_code": "00",
"class_desc": "ALL CLASSES",
"commodity_desc": "CORN",
"congr_district_code": "",
"country_code": "9000",
"country_name": "UNITED STATES",
"county_ansi": "103",
"county_code": "103",
"county_name": "JOHNSON",
"domain_desc": "TOTAL",
"domaincat_desc": "NOT SPECIFIED",
"end_code": "00",
"freq_desc": "ANNUAL",
"group_desc": "FIELD CROPS",
"load_time": "2025-05-06 15:00:00.000",
"location_desc": "IOWA, EAST CENTRAL, JOHNSON",
"prodn_practice_desc": "ALL PRODUCTION PRACTICES",
"reference_period_desc": "YEAR",
"region_desc": "",
"sector_desc": "CROPS",
"short_desc": "CORN, SILAGE - PRODUCTION, MEASURED IN TONS",

"source_desc": "SURVEY",

"state_alpha": "IA",

"state_ansi": "19",

"state_fips_code": "19",

"state_name": "IOWA",

"statisticcat_desc": "PRODUCTION",

"unit_desc": "TONS",

"util_practice_desc": "SILAGE",

"watershed_code": "00000000",

"watershed_desc": "",

"week_ending": "",

"year": 2023,

"zip_5": ""

},

{

"CV (%)": "5.9",

"Value": "25,300",

"agg_level_desc": "COUNTY",

"asd_code": "80",

"asd_desc": "SOUTH CENTRAL",
"begin_code": "00",
"class_desc": "ALL CLASSES",
"commodity_desc": "CORN",
"congr_district_code": "",
"country_code": "9000",
"country_name": "UNITED STATES",
"county_ansi": "135",
"county_code": "135",
"county_name": "MONROE",
"domain_desc": "TOTAL",
"domaincat_desc": "NOT SPECIFIED",
"end_code": "00",
"freq_desc": "ANNUAL",
"group_desc": "FIELD CROPS",
"load_time": "2025-05-06 15:00:00.000",
"location_desc": "IOWA, SOUTH CENTRAL, MONROE",
"prodn_practice_desc": "ALL PRODUCTION PRACTICES",
"reference_period_desc": "YEAR",
"region_desc": "",
"sector_desc": "CROPS",
"short_desc": "CORN, SILAGE - PRODUCTION, MEASURED IN TONS",
"source_desc": "SURVEY",
"state_alpha": "IA",
"state_ansi": "19",
"state_fips_code": "19",
"state_name": "IOWA",
"statisticcat_desc": "PRODUCTION",
"unit_desc": "TONS",
"util_practice_desc": "SILAGE",
"watershed_code": "00000000",
"watershed_desc": "",
"week_ending": "",
"year": 2023,
"zip_5": ""
},
{
"CV (%)": "11.6",
"Value": "50,000",
"agg_level_desc": "COUNTY",
"asd_code": "90",
"asd_desc": "SOUTHEAST",
"begin_code": "00",
"class_desc": "ALL CLASSES",

"commodity_desc": "CORN",

"congr_district_code": "",

"country_code": "9000",

"country_name": "UNITED STATES",

"county_ansi": "051",

"county_code": "051",

"county_name": "DAVIS",

"domain_desc": "TOTAL",

"domaincat_desc": "NOT SPECIFIED",

"end_code": "00",

"freq_desc": "ANNUAL",

"group_desc": "FIELD CROPS",

"load_time": "2025-05-06 15:00:00.000",

"location_desc": "IOWA, SOUTHEAST, DAVIS",

"prodn_practice_desc": "ALL PRODUCTION PRACTICES",

"reference_period_desc": "YEAR",

"region_desc": "",

"sector_desc": "CROPS",

"short_desc": "CORN, SILAGE - PRODUCTION, MEASURED IN TONS",

"source_desc": "SURVEY",
"state_alpha": "IA",

"state_ansi": "19",

"state_fips_code": "19",

"state_name": "IOWA",

"statisticcat_desc": "PRODUCTION",

"unit_desc": "TONS",

"util_practice_desc": "SILAGE",

"watershed_code": "00000000",

"watershed_desc": "",

"week_ending": "",

"year": 2023,

"zip_5": ""

},

{

"CV (%)": "16.2",

"Value": "22,800",

"agg_level_desc": "COUNTY",

"asd_code": "90",

"asd_desc": "SOUTHEAST",

"begin_code": "00",

"class_desc": "ALL CLASSES",
"commodity_desc": "CORN",

"congr_district_code": "",

"country_code": "9000",

"country_name": "UNITED STATES",

"county_ansi": "111",

"county_code": "111",

"county_name": "LEE",

"domain_desc": "TOTAL",

"domaincat_desc": "NOT SPECIFIED",

"end_code": "00",

"freq_desc": "ANNUAL",

"group_desc": "FIELD CROPS",

"load_time": "2025-05-06 15:00:00.000",

"location_desc": "IOWA, SOUTHEAST, LEE",

"prodn_practice_desc": "ALL PRODUCTION PRACTICES",

"reference_period_desc": "YEAR",

"region_desc": "",

"sector_desc": "CROPS",

"short_desc": "CORN, SILAGE - PRODUCTION, MEASURED IN TONS",

"source_desc": "SURVEY",

"state_alpha": "IA",
"state_ansi": "19",

"state_fips_code": "19",

"state_name": "IOWA",

"statisticcat_desc": "PRODUCTION",

"unit_desc": "TONS",

"util_practice_desc": "SILAGE",

"watershed_code": "00000000",

"watershed_desc": "",

"week_ending": "",

"year": 2023,

"zip_5": ""

},

{

"CV (%)": "0.9",

"Value": "4,114,200",

"agg_level_desc": "COUNTY",

"asd_code": "99",

"asd_desc": "",

"begin_code": "00",

"class_desc": "ALL CLASSES",

"commodity_desc": "CORN",
"congr_district_code": "",
"country_code": "9000",
"country_name": "UNITED STATES",
"county_ansi": "",
"county_code": "998",
"county_name": "OTHER COUNTIES",
"domain_desc": "TOTAL",
"domaincat_desc": "NOT SPECIFIED",
"end_code": "00",
"freq_desc": "ANNUAL",
"group_desc": "FIELD CROPS",
"load_time": "2025-05-06 15:00:00.000",
"location_desc": "IOWA, OTHER COUNTIES",
"prodn_practice_desc": "ALL PRODUCTION PRACTICES",
"reference_period_desc": "YEAR",
"region_desc": "",
"sector_desc": "CROPS",
"short_desc": "CORN, SILAGE - PRODUCTION, MEASURED IN TONS",
"source_desc": "SURVEY",
"state_alpha": "IA",
"state_ansi": "19",
"state_fips_code": "19",

"state_name": "IOWA",

"statisticcat_desc": "PRODUCTION",

"unit_desc": "TONS",

"util_practice_desc": "SILAGE",

"watershed_code": "00000000",

"watershed_desc": "",

"week_ending": "",

"year": 2023,

"zip_5": ""

},

{

"CV (%)": "",

"Value": "7,600,000",

"agg_level_desc": "STATE",

"asd_code": "",

"asd_desc": "",

"begin_code": "00",

"class_desc": "ALL CLASSES",

"commodity_desc": "CORN",

"congr_district_code": "",
"country_code": "9000",
"country_name": "UNITED STATES",
"county_ansi": "",
"county_code": "",
"county_name": "",
"domain_desc": "TOTAL",
"domaincat_desc": "NOT SPECIFIED",
"end_code": "00",
"end_code": "00",
"freq_desc": "ANNUAL",
"group_desc": "FIELD CROPS",
"load_time": "2024-09-30 12:00:00.000",
"location_desc": "IOWA",
"prodn_practice_desc": "ALL PRODUCTION PRACTICES",
"reference_period_desc": "YEAR",
"region_desc": "",
"sector_desc": "CROPS",
"end_code": "00",
"freq_desc": "ANNUAL",
"group_desc": "FIELD CROPS",
"load_time": "2024-09-30 12:00:00.000",
"location_desc": "IOWA",
"prodn_practice_desc": "ALL PRODUCTION PRACTICES",
"reference_period_desc": "YEAR",
"region_desc": "",
"end_code": "00",
"freq_desc": "ANNUAL",
"group_desc": "FIELD CROPS",
"load_time": "2024-09-30 12:00:00.000",
"location_desc": "IOWA",
"prodn_practice_desc": "ALL PRODUCTION PRACTICES",
"reference_period_desc": "YEAR",
"end_code": "00",
"freq_desc": "ANNUAL",
"group_desc": "FIELD CROPS",
"load_time": "2024-09-30 12:00:00.000",
"location_desc": "IOWA",
"prodn_practice_desc": "ALL PRODUCTION PRACTICES",
"end_code": "00",
"freq_desc": "ANNUAL",
"group_desc": "FIELD CROPS",
"load_time": "2024-09-30 12:00:00.000",
"location_desc": "IOWA",

"end_code": "00",

"freq_desc": "ANNUAL",

"group_desc": "FIELD CROPS",

"load_time": "2024-09-30 12:00:00.000",

"location_desc": "IOWA",

"end_code": "00",

"freq_desc": "ANNUAL",

"group_desc": "FIELD CROPS",

"load_time": "2024-09-30 12:00:00.000",

"location_desc": "IOWA",

"prodn_practice_desc": "ALL PRODUCTION PRACTICES",

"end_code": "00",

"freq_desc": "ANNUAL",

"group_desc": "FIELD CROPS",

"load_time": "2024-09-30 12:00:00.000",

"location_desc": "IOWA",

"prodn_practice_desc": "ALL PRODUCTION PRACTICES",

"freq_desc": "ANNUAL",

"group_desc": "FIELD CROPS",

"load_time": "2024-09-30 12:00:00.000",
"location_desc": "IOWA",
"prodn_practice_desc": "ALL PRODUCTION PRACTICES",
"load_time": "2024-09-30 12:00:00.000",
"location_desc": "IOWA",
"prodn_practice_desc": "ALL PRODUCTION PRACTICES",
"reference_period_desc": "YEAR",
"region_desc": "",
"sector_desc": "CROPS",
"prodn_practice_desc": "ALL PRODUCTION PRACTICES",
"reference_period_desc": "YEAR",
"region_desc": "",
"sector_desc": "CROPS",
"short_desc": "CORN, SILAGE - PRODUCTION, MEASURED IN TONS",
"reference_period_desc": "YEAR",
"region_desc": "",
"sector_desc": "CROPS",
"short_desc": "CORN, SILAGE - PRODUCTION, MEASURED IN TONS",
"sector_desc": "CROPS",
"short_desc": "CORN, SILAGE - PRODUCTION, MEASURED IN TONS",
"source_desc": "SURVEY",
"state_alpha": "IA",
"state_ansi": "19",

"short_desc": "CORN, SILAGE - PRODUCTION, MEASURED IN TONS",

"source_desc": "SURVEY",

"state_alpha": "IA",

"state_ansi": "19",

"state_fips_code": "19",

"source_desc": "SURVEY",

"state_alpha": "IA",

"state_ansi": "19",

"state_fips_code": "19",

"state_name": "IOWA",

"state_ansi": "19",

"state_fips_code": "19",

"state_name": "IOWA",

"statisticcat_desc": "PRODUCTION",

"state_fips_code": "19",

"state_name": "IOWA",

"statisticcat_desc": "PRODUCTION",

"unit_desc": "TONS",

"state_name": "IOWA",

"statisticcat_desc": "PRODUCTION",
"unit_desc": "TONS",

"util_practice_desc": "SILAGE",

"statisticcat_desc": "PRODUCTION",

"unit_desc": "TONS",

"util_practice_desc": "SILAGE",

"unit_desc": "TONS",

"util_practice_desc": "SILAGE",

"watershed_code": "00000000",

"watershed_desc": "",

"watershed_desc": "",

"week_ending": "",

"year": 2023,

"zip_5": ""

}

]

}
```

Ahora veo el problema. En la respuesta de la API, el campo year viene como un número ("year": 2023) pero nuestro modelo lo está esperando como un string. También hay muchos campos adicionales que no estamos manejando. Voy corregir el modelo:

vamos a usda.go y agregamos `Year                interface{} `json:"year"` // Puede ser string o number`

```
    StateName           string      `json:"state_name"`
    CommodityDesc       string      `json:"commodity_desc"`

    StatisticcatDesc    string      `json:"statisticcat_desc"`

    Value               string      `json:"Value"`

    UnitDesc            string      `json:"unit_desc"`

    ReferencePeriodDesc string      `json:"reference_period_desc"`

    SourceDesc          string      `json:"source_desc"`

    AggLevelDesc        string      `json:"agg_level_desc"`

    UtilPracticeDesc    string      `json:"util_practice_desc"`

    ShortDesc           string      `json:"short_desc"`
```

Ahora necesito actualizar el servicio para manejar el campo Year como interface{} y filtrar solo los datos de nivel estatal y de grano (no silage):

usda_service.go agregamos

```
// Filtrar solo datos de nivel estatal y de grano (no silage)

          if data.AggLevelDesc != "STATE" || data.UtilPracticeDesc != "GRAIN" {

              continue

          }
```

```
            // Convertir year a string
            yearStr := ""
            switch v := data.Year.(type) {
            case string:
                yearStr = v
            case float64:
                yearStr = fmt.Sprintf("%.0f", v)
            case int:
                yearStr = fmt.Sprintf("%d", v)
            default:
                yearStr = fmt.Sprintf("%v", v)
            }
```

```
cornData := models.CornProductionData{
                Year:       yearStr,
                State:      data.StateName,
```

También necesito actualizar la consulta de área cosechada para filtrar correctamente:

```
           // Filtrar solo datos de nivel estatal y de grano
                if area.AggLevelDesc != "STATE" || area.UtilPracticeDesc != "GRAIN" {
                    continue
                }
```

```
// Convertir year del área a string para comparar
areaYearStr := ""
switch v := area.Year.(type) {
case string:
    areaYearStr = v
case float64:
    areaYearStr = fmt.Sprintf("%.0f", v)
case int:
    areaYearStr = fmt.Sprintf("%d", v)
default:
    areaYearStr = fmt.Sprintf("%v", v)
}

if areaYearStr == yearStr && area.StateName == data.StateName {
```

Problema Resuelto:

El error se debía a que la API de USDA Quick Stats devuelve el campo year como un número (2023), pero nuestro modelo lo esperaba como string. También había datos de silage mezclados con datos de grano.

🔧 Correcciones Implementadas:

1. Modelo Actualizado (internal/models/usda.go):

- Cambié Year de string a interface{} para manejar tanto números como strings
- Agregué campos adicionales como AggLevelDesc, UtilPracticeDesc, etc.

1. Servicio Mejorado (internal/services/usda_service.go):

- Agregué filtros para obtener solo datos de nivel estatal (STATE) y de grano (GRAIN)

- Implementé conversión segura del campo year de interface{} a string
- Mejoré el manejo de datos de área cosechada

1. Endpoints Disponibles:

- GET /api/corn-production - Datos de producción de maíz
- GET /api/corn-production/:state - Datos por estado específico
- POST /api/drought-simulation - Simulación de sequía
- GET /simulation - Interfaz web de simulación

Funcionalidades de la Simulación:

- Datos Reales: Consulta la API de USDA para obtener producción real de maíz
- Parámetros Ajustables:
- Año de referencia (2020-2023)
- Estados del cinturón maicero
- Severidad de la sequía (0-100%)
- Área afectada (0-100%)
- Cálculos Económicos:
- Pérdida de producción
- Incremento en precios (elasticidad simplificada)
- Impacto económico total

🚀 Cómo Usar:

1. Acceder a la simulación: http://localhost:8080/simulation
1. Ajustar parámetros en la interfaz web
1. Ejecutar simulación para ver resultados en tiempo real
1. API directa: POST /api/drought-simulation con JSON

Ejemplo de Uso de la API:

```
{

  "year": "2023",

  "states": ["IA", "IL", "NE"],

  "drought_severity": 0.3,

  "affected_area": 0.5

}
```